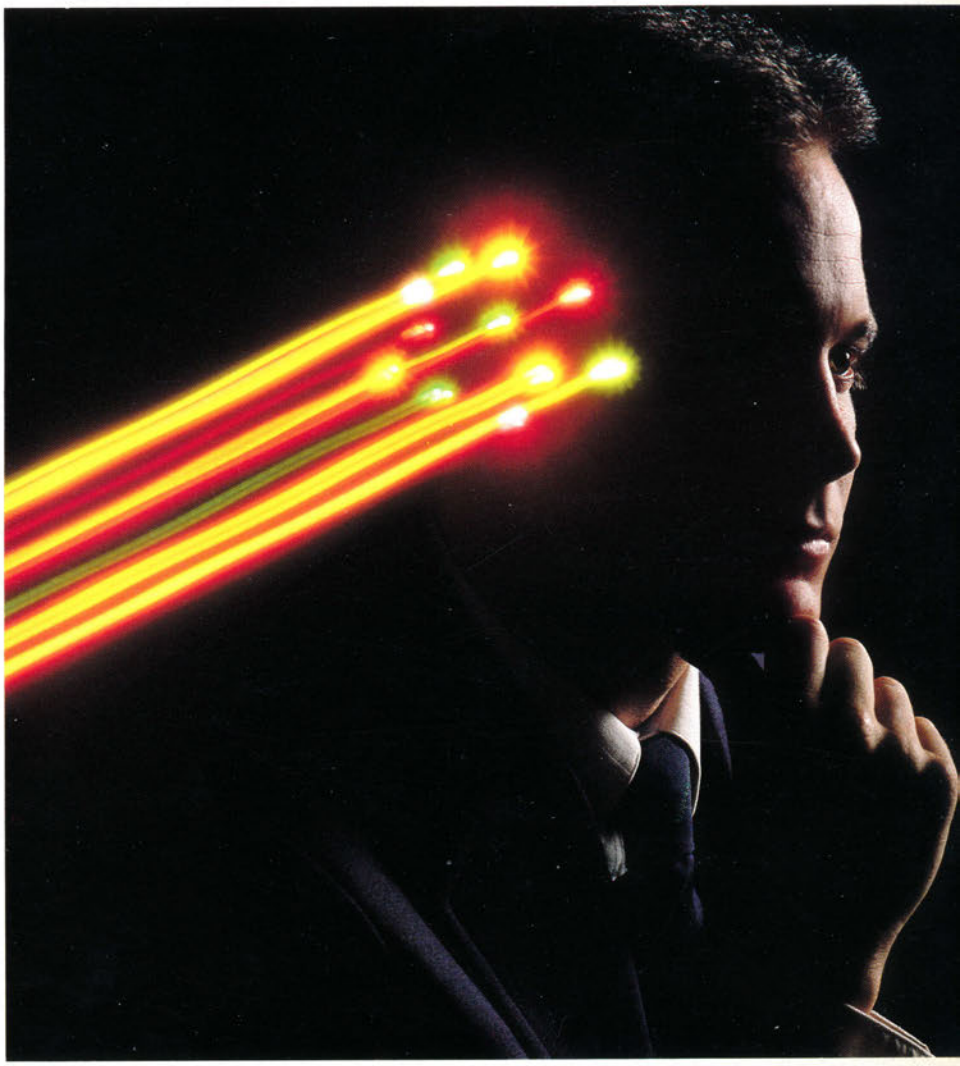
APL ★ PLUS®
Systems

APL
Application
Development
Systems for
Microcomputers



PROBLEM SOLVING AT THE SPEED OF THOUGHT™

STSC

"About five years ago I tried to program the system in BASIC. Hopeless. After six months of starts and restarts, I just kept getting lost in the long listings. It never came together and I think it never would have. When I saw an ad for STSC's APL★PLUS, I sold the Apple®, bought an IBM® PC and finished the program in three weeks. No kidding."

—Michael Garner

```
10 S=0
20 FOR I=1 TO 100
30 INPUT X
40 IF X = 0 GOTO 70
50 S=S+X
60 NEXT I
70 PRINT S/(I-1)
```

BASIC
A program to calculate averages...

```
      REAL X(100)
      READ*,N,(X(I),I=1,N)
      S=0
      DO 10 I=1,N
10    S=S+X(I)
      PRINT *,S/N
      END
```

FORTRAN
Has just shrunk from seven lines...

```
(+/X)÷⍴X←⍞
```

APL
To one.

APL: THE PRODUCTIVITY SOLUTION

Today increasing productivity has become a top priority in business and industry. And that's where APL (A Programming Language) can help. APL's power, conciseness, and problem solving capability spell productivity all around—for the developer, for the integrator and for the end user.

APL symbolizes a whole new way to solve problems. It's an easy-to-use language that enables you to finish projects four to ten times faster than if you had used BASIC, Pascal, or FORTRAN. Because at the heart of APL is a set of easy-to-learn symbols that condense complex operations into a handful of keystrokes.

Working in Shorthand is Faster and Smarter

APL's concise notation and robust syntax eliminate the need for writing routines that are incidental to the task, such as sorting, matrix inversions, string-searching, table manipulation and more. These features are part of the language, so you can concentrate on the problem and not worry about the implementation.

Forget all the DO, FOR, and WHILE loops you need to sort numbers or names when using FORTRAN, BASIC, or Pascal. With APL one symbol does it all.

For example, let X be a list of numbers:

```
X ← 4 1 8 6 2 6 51 8
```

To sort them in ascending order, just type:

```
X[⍋X]
```

You get:

```
1 2 4 5 6 8 8 51
```

And alphabetical sorting is just as simple!

To get the maximum value of a list of numbers using BASIC you could use the following code:

```
10 MAX = -100
20 FOR I = 1 TO 5
30 INPUT X
40 IF X<MAX GOTO 60
50 MAX = X
60 NEXT I
70 PRINT MAX
```

Compare this to APL's maximum value command:

```
⌈/X
```

Minimum value command:

```
⌊/X
```

And the command to total:

```
+/X
```

In APL, if you need to divide two strings of numbers, all you need is the ÷ symbol!

For example, let REV represent the revenue from the sale of nuts, bolts, and screws, and QUANT be the amount sold.

You type:

```
REV ← 584.60 613.11 32.10
QUANT ← 158 321 15
```

One simple APL statement gives you the average selling price of each item:

```
REV ÷ QUANT
3.7 1.91 2.14
```

"I wouldn't want APL's ability to write commercial applications to overshadow its great friendliness and fun. APL is a language you can think in."

—Mark Wilson, M.D.

APL★PLUS: THE APPLICATION DEVELOPMENT SOLUTION

APL★PLUS Systems are tailored for productivity. More than just powerful APL language interpreters, APL★PLUS Systems combine the power of the APL language with problem solving tools one typically finds in fourth generation application development systems. The power of the language, coupled with a well integrated application development environment, make solving problems at the speed of thought a reality.

Full-Screen Management

Full-Screen features provide powerful display and editing facilities. Window-oriented screen input/output functions make it easy to construct input forms or pop-up menus.

Also available in many of the APL★PLUS products is a powerful word processing-style editor for rapid modification of both programs and character data. And, a numeric editor provides users with a facility to modify numeric data with the same ease and flexibility as spreadsheets.

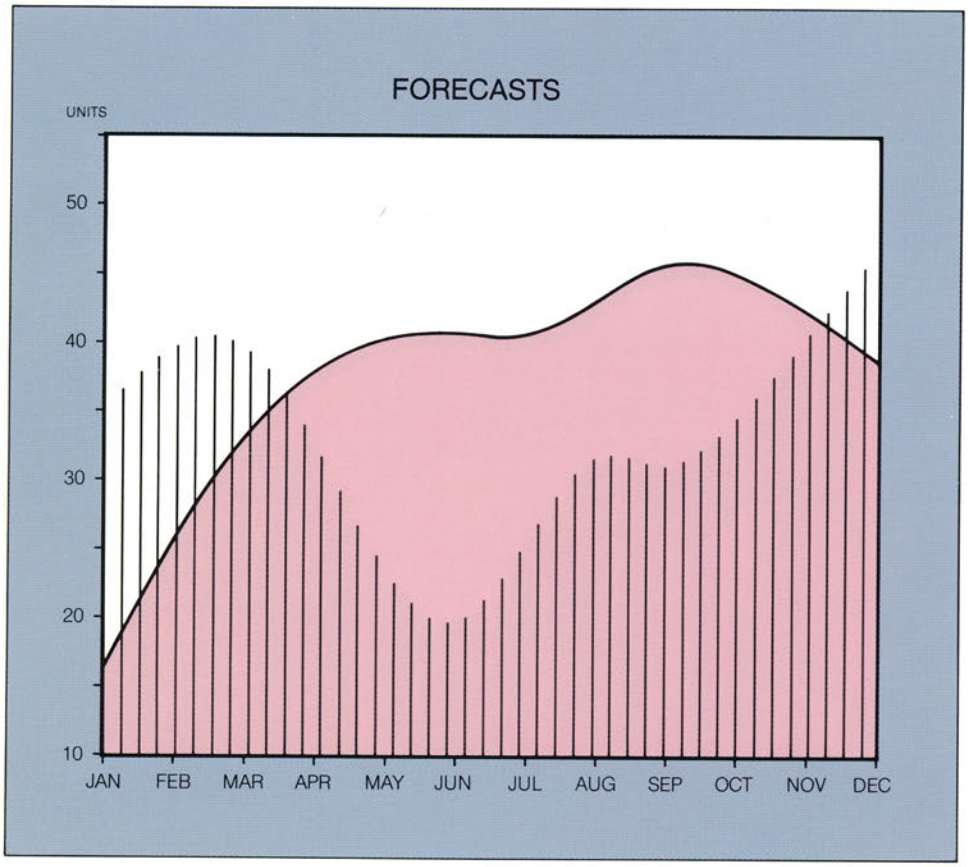
File Systems

A powerful file system makes data management easy. Built-in file functions permit APL arrays of any type or size to be efficiently stored and retrieved from files without requiring the user to know extraneous details about file organization. Multi-user versions of the APL★PLUS System provide extensive file-sharing facilities enabling multiple users to access and even update a file simultaneously.

Communications

A powerful communications mode enables the computer to act as either an APL or ASCII terminal to other computers. Programs and system primitives are provided to support the transfer of programs and data. The communications feature can be used to drive pen plotters or other peripheral devices such as bar-code scanners.

David Mangini of the Southern New England Telephone Company, uses the APL★PLUS PC System to monitor the operation of a cellular mobile telephone network. "The APL★PLUS System was the only intelligent application system that provided all of the communications, graphics and computational facilities needed in one package."



Report Formatting

A general purpose formatting capability provides a simple, yet powerful method of producing attractive reports. STSC's well-known report formatting facility combines the best of FORTRAN and COBOL picture editing to make any data formatting easy.

SALES REPORT

	NUTS	BOLTS	SCREWS	TOTAL
QUANTITY	158	321	15	494
REVENUE	$584.60	$613.11	$32.10	$1,229.81
AVG. SALES PRICE	$3.70	$1.91	$2.14	$2.49

With APL's report formatting capabilities, you can produce the report exactly the way you want.

Error Handling

Error-handling features mean that applications can act friendly, even when inexperienced users make mistakes. STSC's exception-handling features allow applications to react automatically to errors and exceptions that occur during program execution, and take corrective action, or simply keep the program from terminating. Exception-handling can also improve the security of applications, protecting programs and data from accidental or malicious interference.

Help Facility

A HELP Facility provides users with a means to display information describing how to use the features of the APL★PLUS System. With this feature, application developers can create user documentation for their custom applications.

Integrated Applications

The APL★PLUS Systems can be integrated with other applications you have already developed, or can be used to integrate applications that currently do not "talk" to one another. The powerful communications facilities and the ability to access any files, in combination with the ability to execute non-APL programs from APL, provide a unique integration mechanism.

"Everything I had written [for the mainframe computer] was completely compatible. It was as if I had never left the mainframe system."

—Lee Argush

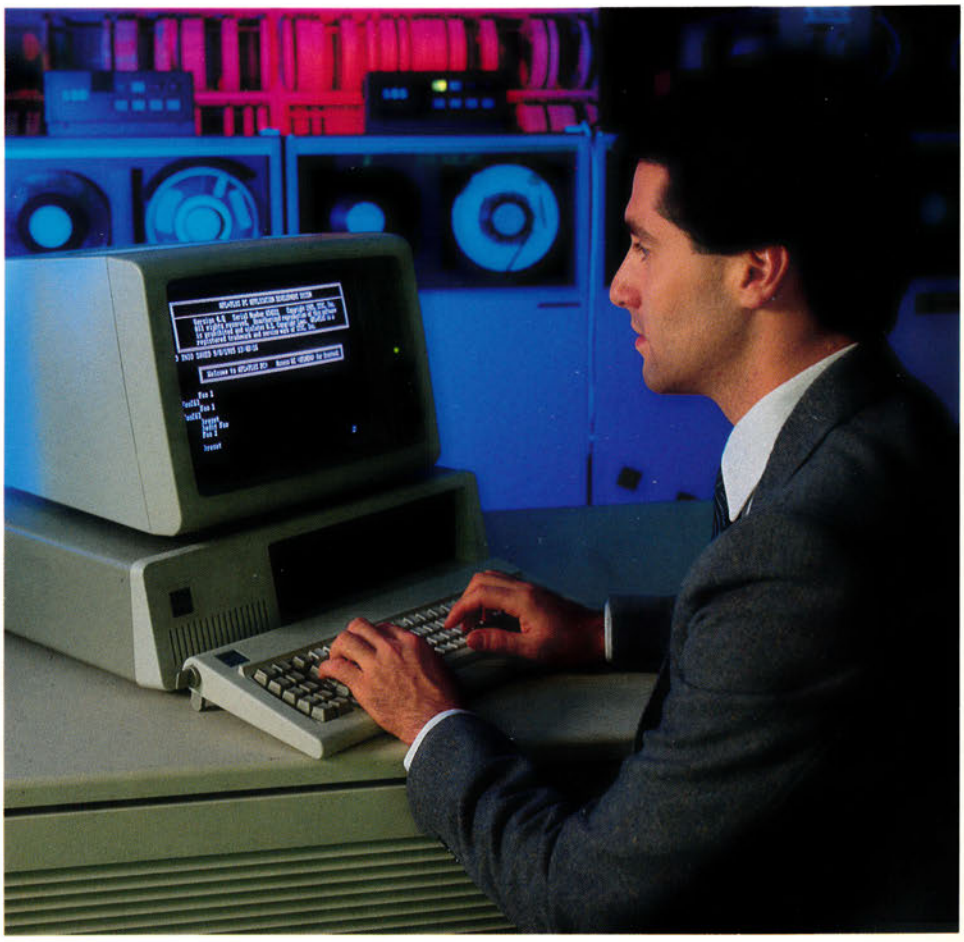
Run-Time Program
The APL★PLUS Run Time option is available to application developers who want to distribute APL-based software. With the Run-Time option, the application developers use the powerful features of the APL★PLUS System and produce software that does not require the user to purchase a complete APL★PLUS System.

Portability
APL★PLUS Systems operate on the entire range of computers, from desktops to the largest of mainframes. So, as your needs change, your APL★PLUS application can be moved to different computers—with little or no conversion.

STSC's approach to language processors provides low cost portability from one computer architecture to another. Ongoing research and development efforts are dedicated to enhancing and extending APL to run in a variety of environments in the most productive and powerful manner possible.

Performance
STSC's APL★PLUS Systems outperform not only competitive APL systems, but compiled languages and other application development systems.

Our Research and Development team is discovering new ways to maximize speed through partial compilation techniques and effective use of floating point hardware.



Quality Products and Service
STSC's APL★PLUS Systems are proven. Their features have been used for over a decade in computer timesharing environments and application development. STSC makes regular updates to the APL★PLUS Systems. Updates provide new features and changes, incorporating your comments and suggestions, and adding new features to keep the software current with the latest technology available.

A telephone HELP-line puts the user in contact with an experienced staff to provide consultation. And, our regularly published newsletters describe new products and features and provide useful programming tips.

"I chose APL★PLUS PC over other APL interpreters for the PC because of its many advanced features. In addition, the telephone HELP-line provides valuable support for the product."

—Mark Schroeder

Easy to Learn and Use

APL is easier to learn than BASIC. The APL★PLUS Systems come with an APL tutorial to make learning APL easy. Comprehensive documentation and reference manuals provide all the information necessary for effective use of the software. Within a matter of days you will be solving complex problems simply and easily.

APL is especially fun to use. Ask anyone who uses it! With APL you will enjoy yourself and be productive at the same time.

APL in Education

STSC's APL★PLUS software is presently in use at over 400 of the undergraduate universities in the United States—ensuring the future of APL in the businesses of tomorrow.

In the educational environment, APL★PLUS Systems are not limited to programming instruction. Students with little or no prior programming experience use APL to solve complex problems in the areas of mathematics, science, and business.

STSC, the Company

STSC, Inc. provides productivity tools and decision support software internationally. STSC has offices in the United States and licenses software through distributors worldwide.

As an industry leader in the development of the APL programming language, STSC delivers APL-based software and services in the microcomputer and mainframe environments, and publishes software written by independent developers.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852

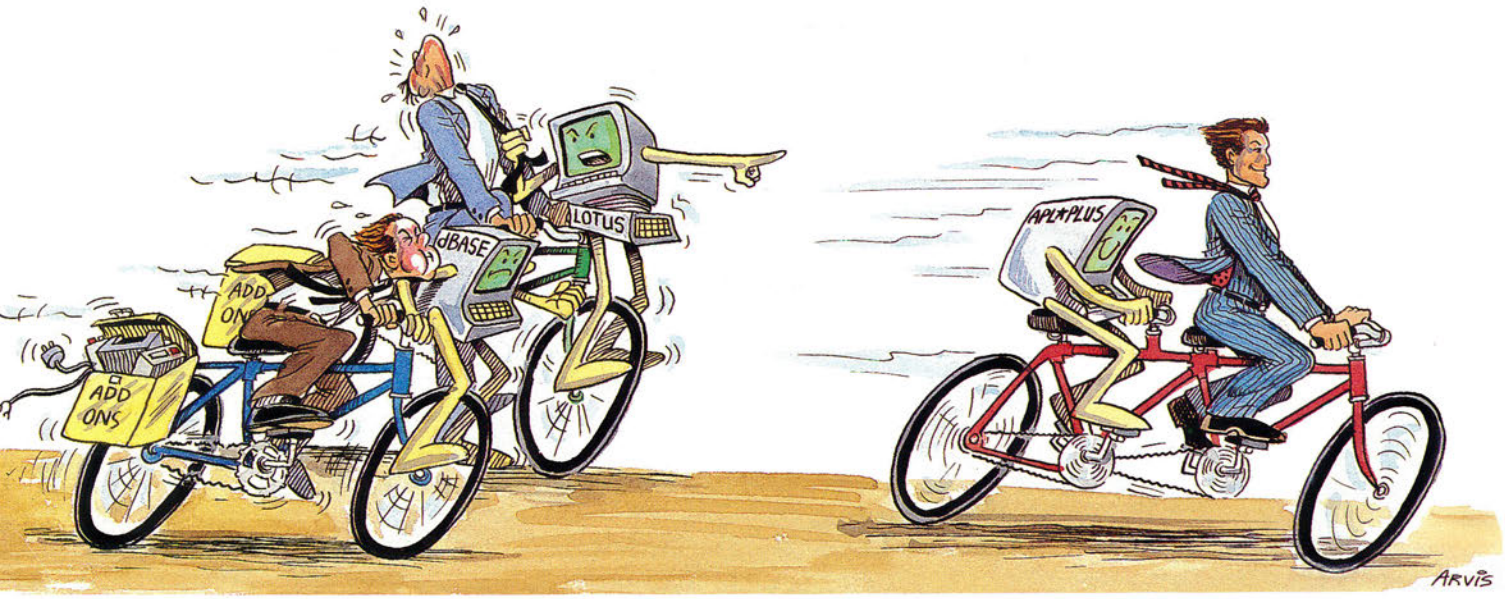
A PLUS★WARE™ PRODUCT

APL★PLUS is a registered trademark and
PLUS★WARE is a trademark of STSC, Inc.
Apple, and IBM are registered trademarks of
Apple Computer, Inc., and International Business
Machines, respectively.

Copyright © 1985, STSC, Inc.
Printed in the U.S.A.
CC210-1185

A PERSONAL LANGUAGE

THE APL★PLUS® SYSTEM AND YOU



STSC

LOTUS®, dBASE®, AND MORE...

Lotus, dBASE, and other popular business software provide the basic tools necessary to survive in today's business environment. But while these business packages provide us with an easy way to set up tabular data, enter numbers, and compute simple model calculations, attempts to do more are often complicated and time-consuming.

That's where APL★PLUS Systems come in. APL★PLUS Systems' unique analytical powers give you the ability to solve complex business problems that packages like Lotus and dBASE simply are not designed to handle. With powerful built-in functions and concise code, you can solve complex problems quickly and easily.

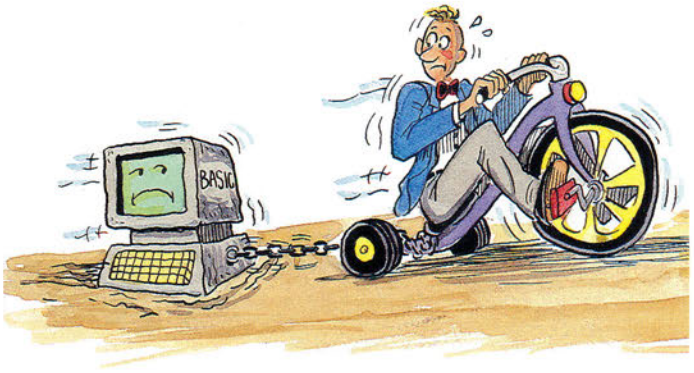
INTERFACING WITH LOTUS AND dBASE FILES

With the APL★PLUS System Spreadsheet Manager you can import data from existing Lotus or dBASE files, perform your calculations, and return the results to their original format. Moving data in and out of these packages is simple with the Spreadsheet Manager and there is no need to alter your existing Lotus or dBASE data to get it into the right format as the Spreadsheet Manager does this for you.

BEYOND COMMON BUSINESS TOOLS

What sets APL★PLUS Systems apart from other languages and business tools is its natural mathematical orientation and concise notation. Concise, powerful, and best of all, easy to learn and use. That's because at the heart of APL is a set of easy-to-learn symbols that condense complex operations into a handful of keystrokes.

Rather than mold calculations and algorithms to a programming language style, APL expands upon familiar algebraic notation, thus viewing the problem the



"We started with advanced VisiCalc but it quickly became obvious our requirements were far beyond a simple spreadsheet application. The APL★PLUS System allows us to manage one quarter million numbers in 30 files on the micro. Any company using a broker distribution network must face the same challenge. Think of it—one quarter million numbers and no math errors."

Bruce E. Holtermann, Controller
Virginia Capes Seafood, Inc.

"A spreadsheet (1-2-3) template was considered, but this did not provide enough error control or flexibility.... APL is an excellent language for handling the matrix calculations which were required in order to quote a choice of coverage options."

Howard H. Friedman, Actuary
Medical Mutual Liability Insurance

way people view the problem. By adhering to a common mathematical notation, problems and calculations are:

- easier to write
- shorter and easier to follow
- faster to modify, and
- mostly loop-free.

Unlike common compiled programming languages, APL is an interpreted language, so results are immediate. Programs can be developed and debugged several times faster because changes can be made during the testing phase without reverting back to the compile-link-execute stage.

SOLVE PROBLEMS EASIER, FASTER WITH APL★PLUS SYSTEMS

Compare solutions to the following typical problem. In this example, we compute withholding taxes based on various tax rates.

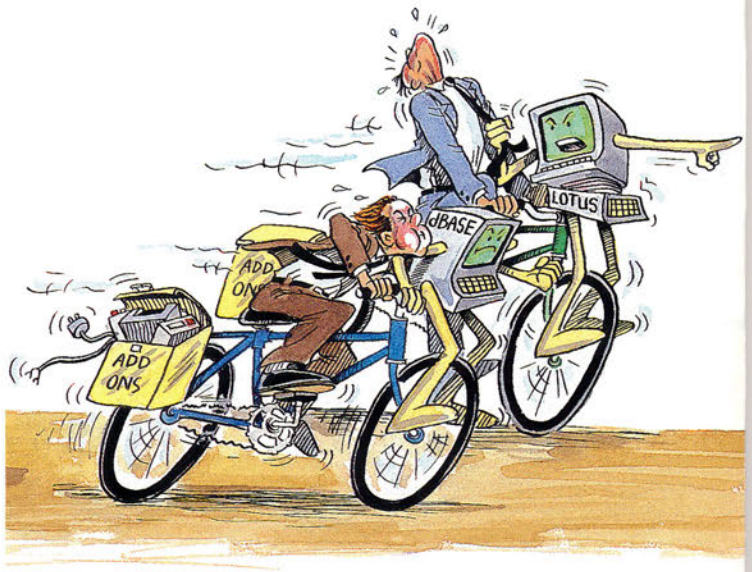
WITHHOLDING TAX TABLE*

MONTHLY PAY RANGE	WITHHOLDING TAX	
	BASE AMT.	RATE
UP TO $199	0	
$199 UP TO $295	...	11% OVER $199
$295 UP TO $382	$126.50	+ 12% OVER $295
$382 UP TO $563	$251.30	+ 14% OVER $382
$563 AND OVER	$556.50	+ 15% OVER $563

*ADAPTED FROM 1985 FEDERAL INCOME TAX TABLE FOR SINGLE WORKERS

A POSSIBLE dBASE SOLUTION...

```
WITHHOLD=0
IF WAGE>199
 WITHHOLD=.11*(WAGE-199)
  IF WAGE>295
   WITHHOLD=126.5+.12*(WAGE-295)
    IF WAGE>382
     WITHHOLD=251.3+.14*(WAGE-382)
      IF WAGE>563
       WITHHOLD=556.5+.15*(WAGE-563)
      ENDIF
    ENDIF
  ENDIF
ENDIF
```

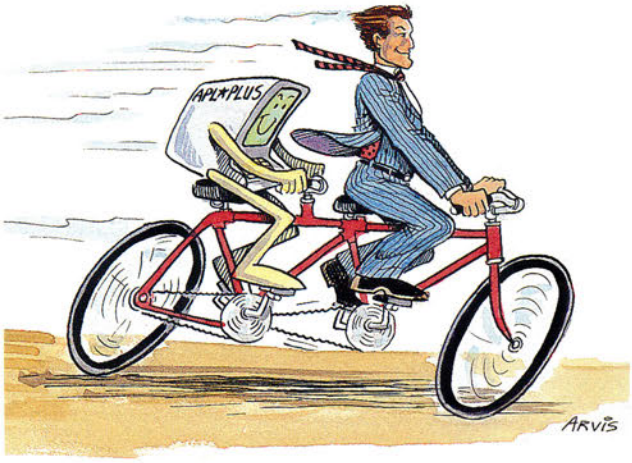


"The indexing and subscripting of so many matrices, many of different dimension, would have been extremely tedious (and error prone) in BASIC or FORTRAN. The analytic power of dBASE is totally inadequate to this task. Questionnaire data had been accumulated in dBASE II files. STSC's native file system made it easy to read these files and convert them to APL variables."

Dick Holt
Director of Computer Applications
HRH Systems, Inc.

"The language is especially well suited for mathematical programming and research needs in which programming must often be modified or revised to suit a unique project. I chose APL★PLUS... because of its many advanced features. Programming in APL was straightforward. Transformations of data and processing results of programs is easy in APL."

Mark Schroder
Chicago Board of Trade



A POSSIBLE LOTUS SOLUTION...

```
(WAGE-@VLOOKUP(WAGE,$TAXTABLE,0))
*VLOOKUP(WAGE,$TAXTABLE,2)
+@VLOOKUP(WAGE,$TAXTABLE,1)
```

A POSSIBLE BASIC SOLUTION...

```
510  WITHHOLD=0
520  IF WAGE>199 THEN
       WITHHOLD=.11*WAGE-199
530  IF WAGE>295 THEN
       WITHHOLD=126.5+.12*(WAGE-295)
540  IF WAGE>382 THEN
       WITHHOLD=251.3+.14*(WAGE-382)
550  IF WAGE>563 THEN
       WITHHOLD=556.5+.15*(WAGE-563)
```

AND, IN APL...

```
I←+/WAGE≥RANGE
WITHHOLD←(BASE+RATE×(WAGE−RANGE))[I]
```

Imagine the effort required in other systems to find the withholding tax for many employees from tables with many tax brackets. In APL, only a slight modification is needed to extend the solution to any number of employees or tax brackets.

```
I←+/WAGE°.≥RANGE
```

Another advantage of APL is the flexibility to ask many questions after the initial calculation is done:

- What is the largest withholding tax?
  `⌈/WITHHOLD`
- How many employees have withholdings in excess of $250 per month?
  `+/(WITHHOLD>250)`
- What is the average withholding tax?
  `(+/WITHHOLD)÷⍴WITHHOLD`
- What is each employee's FICA tax?
  `.0715×3500⌊WAGE`

"APL★PLUS permits radical changes to be introduced quickly and easily; no other software currently offers this flexibility at this speed. By basing our forecasting capabilities in APL★PLUS I have complete control. I cannot conceive of any other language offering this degree of control and of anything other than APL★PLUS doing it as well."

Richard J. Goettle
Cambridge Planning & Analytics, Inc.

".... none of the other software that I am aware of can give the least-squares solution of an overdetermined system of linear equations with a single keystroke, as APL can."

Charles Libove
Professor, Mechanical and Aerospace Engineering
Syracuse University

A FULL-RANGE OF APL★PLUS PRODUCTS AND SERVICES

STSC's technology, based on 17 years of experience and development of the APL programming language, provides customers with the most advanced APL language systems available today—APL★PLUS Systems.

APL★PLUS Systems operate on the entire range of computers—PC's and PC compatibles, UNIX machines, and mainframes. So, as your needs change, your APL★PLUS applications can be moved to different computers—with little or no conversion. And, our comprehensive APL★PLUS line also includes a worldwide computer timesharing service, giving you more options for your data handling needs.

Our commitment to APL★PLUS System technology goes far beyond the development of our system software with professional services that include consulting, custom design and programming, technical support, and customer training.

STSC's consultants provide specific application skills, customizing applications to run for your specific needs. To help you further develop the APL skills you need in your particular business, STSC's Educational Division and APL Press book publishing division offer a wide range of courses, seminars, books, and tutorials.

So, if you've reached the limits of your present software packages, put the power, speed and flexibility of an APL★PLUS System or service to work for you. See your local dealer or call 1-800-592-0050. In Maryland, or Canada, call 301-984-5123.

"The real strength of interpreters, and of APL in particular, is with problems that aren't yet fully defined, problems that are still being worked out. I needed an understanding of anesthetic uptake, not of computer programming. APL filled that need by providing an environment where I could play around with a lot of disconnected concepts that I hadn't yet put together, a place where I could try out new ideas quickly. For this, APL shines. I wouldn't want APL's ability to write commercial applications to overshadow its great friendliness and fun. APL is a language you can think in."

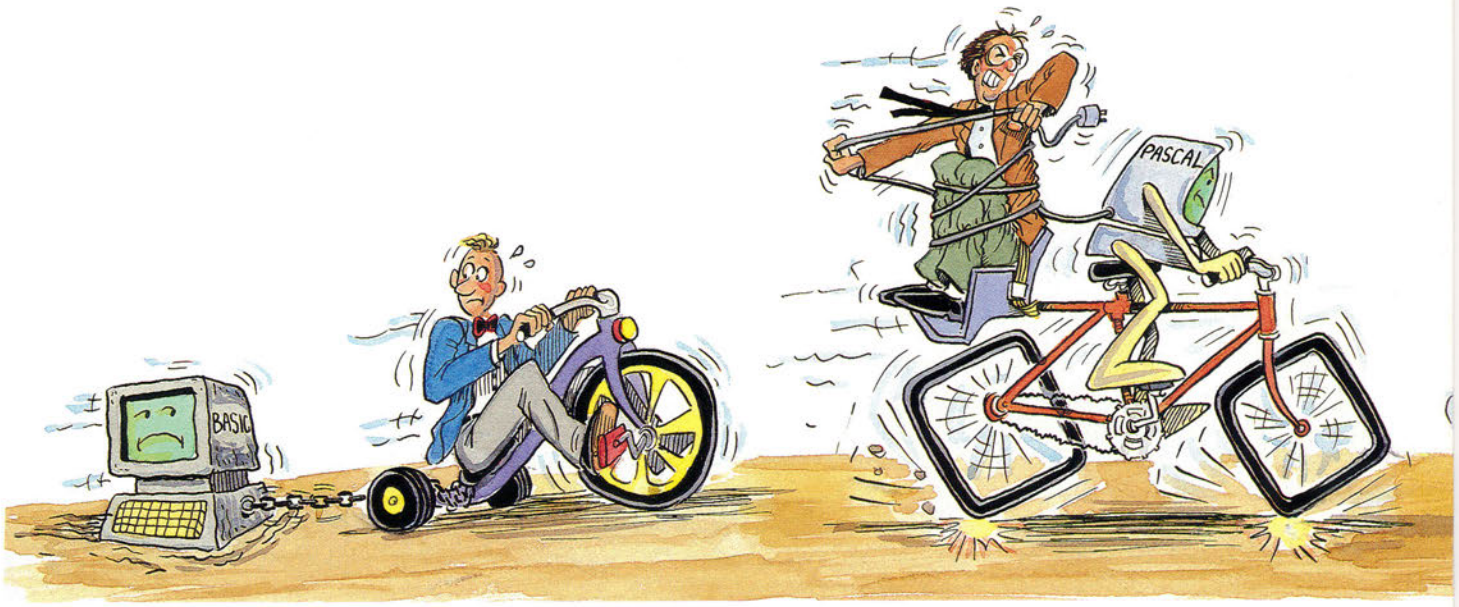
Mark Wilson, MD.
Resident Physician—Anesthesiology
University of Colorado Health Sciences Center

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, MD 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

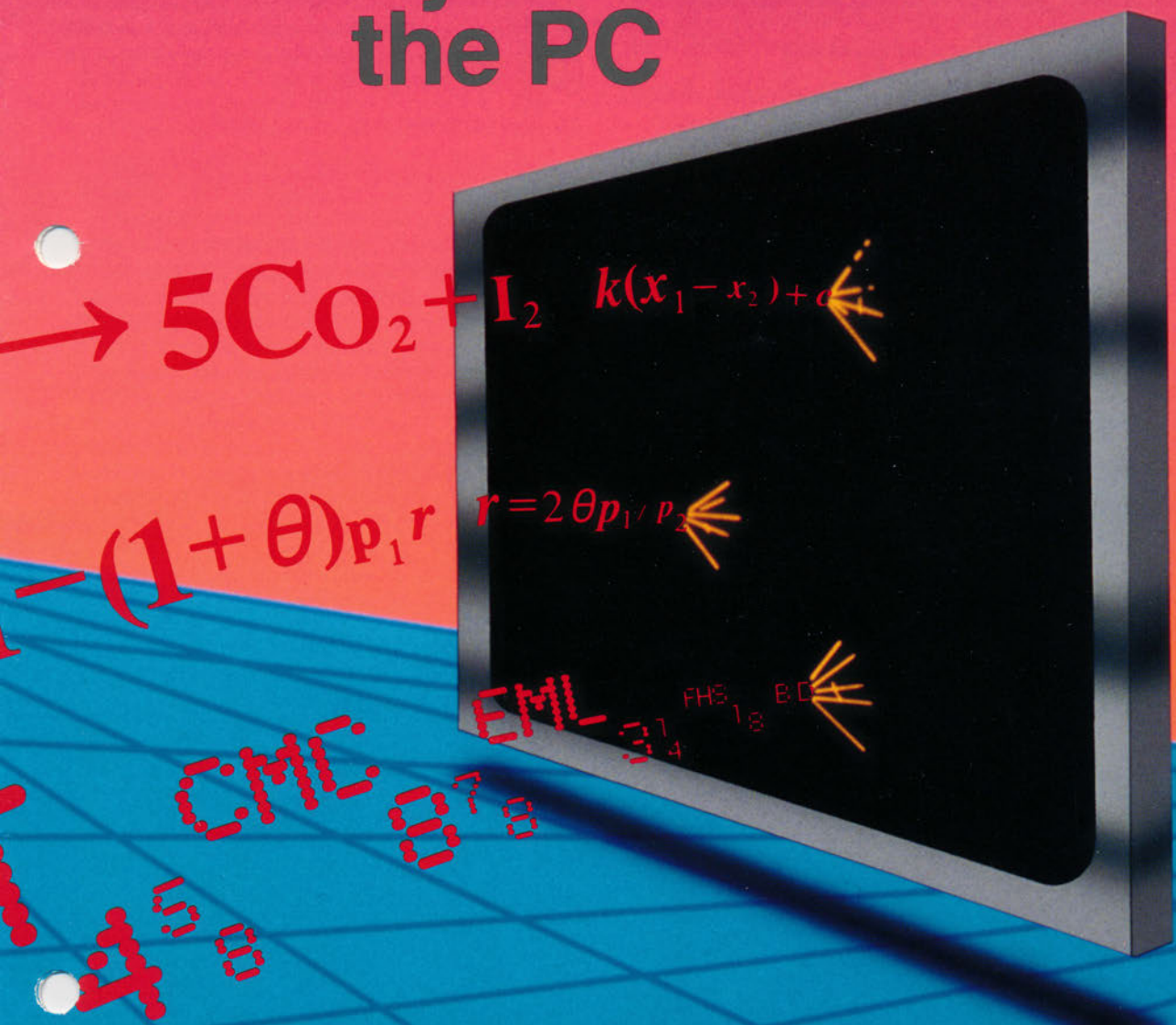
APL★PLUS is a trademark and service mark of STSC, Inc.
PLUS★WARE is a trademark of STSC, Inc. Lotus and dBASE are trademarks of Lotus Development Corporation and Ashton-Tate, respectively.

Copyright © 1986, STSC, Inc.
Printed in the U.S.A.
CC213-0886



A PLUS★WARE™ PRODUCT

APL★PLUS® System for the PC



STSC

THE APL★PLUS SYSTEM

"The language is especially well-suited for mathematical programming and research needs in which programs must often be modified or revised to suit a unique project. I chose APL★PLUS because of its many advanced features. Programming in APL was straightforward. Transforming data and processing results of programs are easy in APL."

Mark Schroder

Chicago Board of Trade

The APL★PLUS System for the PC

The APL★PLUS System for the PC has been tailored to the personal computer environment to give you access to all of the PC's capabilities. You have complete control over the screen and keyboard, full support for graphics, direct access to the DOS operating system, and connectivity to other environments.

Full-screen Management

Full-screen features provide powerful display and editing facilities. Window-oriented screen input/output functions make it easy to construct input forms or pop-up menus.

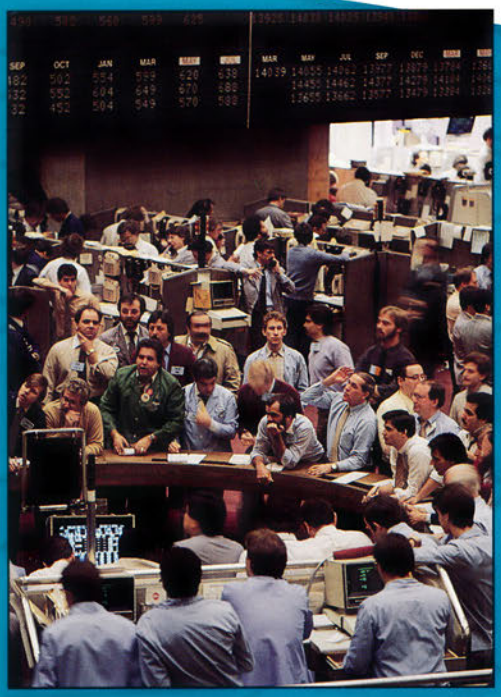
A powerful word processing-style editor allows quick modification of programs and character data. And a numeric editor provides a facility to modify numeric data with the same ease and flexibility as spreadsheets.

File Systems

A powerful file system makes data management easy. Built-in file functions permit APL arrays of any type or size to be efficiently stored and retrieved from files without requiring you to know extraneous details about file organization. With PCs connected by local area networks, the APL★PLUS System provides extensive file sharing facilities enabling multiple users to access and even update a file simultaneously.

Communications Versatility

A powerful communications mode enables the computer to act as either an APL or ASCII terminal to other computers. An optional tool provides 3270-like mainframe communications. Programs and system primitives are provided to support the transfer of programs and data. The communications feature can be used to drive printers, plotters, or other peripheral devices such as bar-code scanners.



Report Formatting

A general purpose formatting capability provides a simple, yet powerful method of producing attractive reports. STSC's well-known report formatting facility uses a concise notation to describe reports — one statement can generate an entire report!

Error Handling

Error-handling features mean that applications can act friendly, even when mistakes are made. STSC's exception-handling features allow applications to react automatically to errors and exceptions that occur during program execution, and take corrective action, or simply keep the program from terminating. Exception-handling can also improve the security of applications, protecting programs and data from accidental or malicious interference.

Graphics

Sophisticated application design often requires graphic presentation of data. The APL★PLUS System for the PC contains built-in functions for screen graphics, as well as an interface to the industry standard GSS*CGI graphics drivers, so you can easily incorporate high-quality graphics into your applications.

Integrated Applications

The APL★PLUS System for the PC can be integrated with other applications you have already developed, or can be used to integrate applications that currently do not "talk" to one another. The powerful communications facilities and the ability to access any files, in combination with the ability to execute non-APL programs from APL, provide a unique integration mechanism.

Runtime Program

The APL★PLUS Runtime option is available to application developers who want to write and distribute APL-based software. With the Runtime option, the application developer uses the powerful features of APL★PLUS and produces application software that does not require the end user to purchase a separate system.

Portability

APL★PLUS systems are available for PCs, workstations, minicomputers, and the largest of mainframes. So, as your needs change, your APL★PLUS application can be moved to different computers — with little or no conversion.

"A spreadsheet (1-2-3) template was considered, but this did not provide enough error control or flexibility. . . APL is an excellent language for handling the matrix calculations which were required in order to quote a choice of coverage options."

Howard H. Friedman, Actuary

Medical Mutual Liability Insurance

STSC's approach to language processors provides low cost portability from one computer architecture to another. Ongoing research and development efforts are dedicated to enhancing APL to run in a variety of environments in the most productive and powerful manner possible.

Performance

STSC's APL★PLUS products not only outperform other APL systems, but often surpass compiled languages and other application development systems. STSC's research and development team is constantly working to improve the speed of APL.

Quality Products and Services

Many features of STSC's APL★PLUS products have been in use for nearly 20 years, beginning in computer timesharing and application development environments. STSC regularly updates all versions of its APL★PLUS products with new features and changes, incorporating your comments and suggestions.

Our telephone HelpLine puts the user in contact with an experienced staff to answer client questions. And, our quarterly newsletter describes applications and new products and provides useful programming tips.

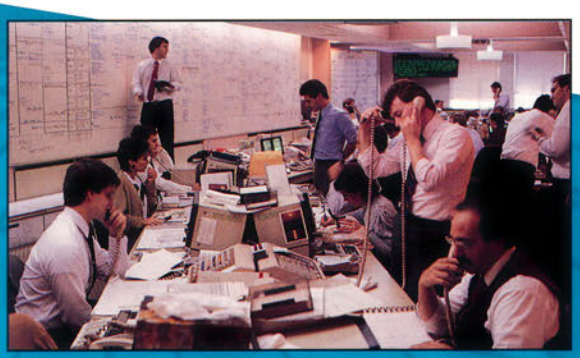
Easy To Learn and Use

The APL★PLUS System for the PC comes with an APL tutorial to make learning APL easy. Comprehensive user's guides and reference manuals provide all the information necessary for effectively using the software. Within a matter of days you will be solving complex problems simply and easily. STSC also provides a full complement of APL courses at sites across the country to train both beginners and advanced APL users.

APL is especially fun to use. Ask anyone who uses it! The productivity and ease with which your ideas turn into working applications make using APL a pleasure.

APL in Education

STSC's APL★PLUS software is in use at over 500 undergraduate and graduate colleges and universities around the world — ensuring the future of APL in the businesses of tomorrow.



"The real strength of interpreters, and of APL in particular, is with problems that aren't yet fully defined; problems that are still being worked out. I need an understanding of anesthetic uptake, not of computer programming. APL filled that need. . ."

Mark Wilson, M.D.

Resident Physician-Anesthesiology

University of Colorado Health Sciences Center

STSC

APL: The Productivity Solution

Today, increasing productivity has become a top priority in businesses around the world. That's where APL shines! APL's power, concise code, and dynamic problem-solving capability spell productivity.

APL is an easy-to-use language packaged in a user-friendly computing environment. At the simplest level, APL is an interactive tool for analyzing numeric data. However, with APL it is easy to go from interactive numeric calculations, to simple programs, to developing complete end-user application systems. You don't have to master APL to see tremendous gains in productivity, but the more you learn the greater the productivity improvement.

An experienced APL user can build applications as sophisticated as with any programming language in much less time. Numerous studies over time have shown that APL lets you finish projects up to 10 times faster than with conventional programming languages like C, Fortran, or Pascal. And the flexibility, large capacity, and speed of STSC's APL★PLUS System make it ideal for problems that aren't suitable for spreadsheets, database managers, and other high level software tools.

Because APL handles tables and large multidimensional arrays of data so easily, it is no surprise that APL has become the "tool of choice" for anyone having complex numeric problems to solve. Thousands of professionals — financial analysts, actuaries, researchers, statisticians, and engineers, to name a few — depend on APL to help them meet the rapidly changing requirements of their jobs in the most effective, productive way possible.

Work Faster and Smarter

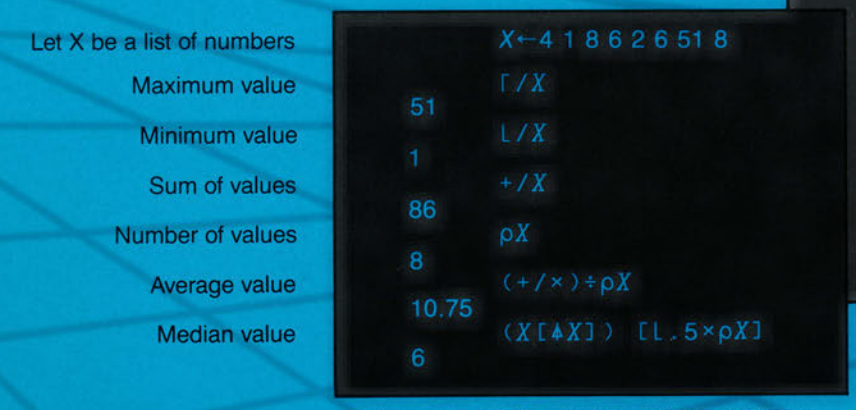
APL programs are concise and powerful. The built-in operations of APL work on entire arrays of data instead of forcing you to deal with one number or one character at a time. APL promotes array-oriented thinking, making it easy to perform big calculations with short APL statements. APL notation is as concise as mathematical notation — most operations can be represented with a single symbol, so a few lines of APL are equivalent to pages of code in a conventional language.

Moreover, APL lets you focus your thinking on the problem being solved, not on the details of programming. One symbol in APL can take the place of a DO, FOR, or WHILE loop in a language like Fortran or C. There's no need to declare variables — APL automatically selects the correct datatype and size to fit your data.

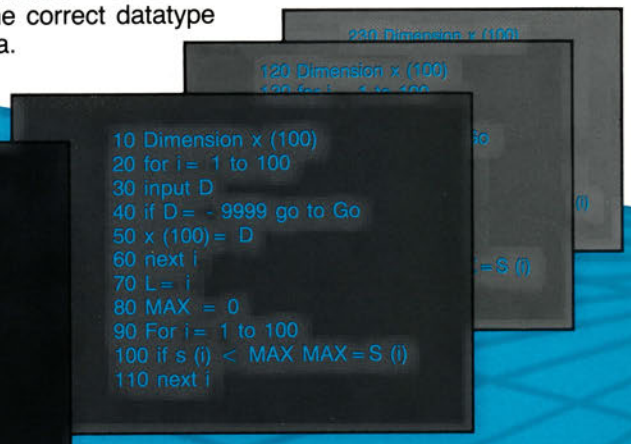
"APL★PLUS permits radical changes to be introduced quickly and easily; no other software currently offers this flexibility at this speed. By basing our forecasting capabilities in APL★PLUS, I have complete control. I cannot conceive of any other language offering this degree of control and of anything other than APL★PLUS doing it as well."

Richard J. Goettle

Cambridge Planning and Analytics, Inc.



Deriving basic statistics about a set of numbers is easy and interactive with APL. A few keystrokes suffice for simple statistical analysis.


10 Dimension x (100)
20 for i = 1 to 100
30 input D
40 if D = - 9999 go to Go
50 x (100) = D
60 next i
70 L = i
80 MAX = 0
90 For i = 1 to 100
100 if s (i) < MAX MAX = S (i)
110 next i

The equivalent calculations in Basic require programs running to several pages.

STSC, The Company

In 1982, STSC introduced the first commercially successful APL for the IBM PC. Since then, the APL★PLUS System for the PC has become the most popular version of the APL language, with more than 30,000 systems shipped worldwide. STSC has more than 19 years experience in the computer software and services industry and is the leader in the development of APL programming technology.

Today, STSC provides a family of compatible versions of its APL★PLUS products on a wide range of machines, from IBM mainframes to DEC VAX minicomputers and UNIX workstations, as well as personal computers. In addition to supporting more than 50,000 APL software users worldwide, STSC offers APL-based remote computing services, APL systems and applications consulting, and APL training.

APL software complements the company's comprehensive line of business analysis software products which assist in the quantitative analysis of statistical, geographical, and logistical data. These products incorporate high-quality, fully integrated graphics to most effectively facilitate these data analysis and decision-making processes.

STSC

2115 East Jefferson Street
Rockville, Maryland 20852
800-592-0050
301-984-5123 (in Maryland or outside the U.S.)
Telex: 898085 Fax: 301-984-5094

APL★PLUS is a registered trademark of STSC, Inc.
Copyright© 1988, STSC, Inc.
Printed in U.S.A.

STSC, THE COMPANY

STSC, Inc. was founded in 1969 to provide remote computing services based on an enhanced version of the APL programming language. Since that time STSC has become the largest U.S. supplier of APL-based products with APL★PLUS® System users worldwide. Our development of proprietary extensions to the APL language has resulted in the unique and powerful APL★PLUS Application Development System. In 1979 STSC expanded its service offering to include licensed system software and application software.

In 1982 STSC entered the microcomputer marketplace with an implementation of the APL★PLUS System for the microcomputer. In 1984 the software publishing business grew to include software products authored by independent developers. STSC acquired APL Press of Palo Alto, California in 1985, further extending its publishing activities.

STSC's large staff of APL professionals results in a unique concentration of APL talent and expertise. In addition to APL, we have many years of experience in other popular languages such as FORTRAN, Assembler, PL1, PASCAL, C, BASIC, and COBOL. And, our familiarity with operating systems includes the VM, MVS, UNIX™, PC DOS, and MS-DOS® environments. With this diverse experience, we have the knowledge to interface to your particular inhouse language requirements.

Headquartered in Rockville, Maryland, STSC provides its broad range of telecommunications and software services internationally through branch offices and a worldwide dealer base.

APL★PLUS and MS-DOS are registered trademarks of STSC, Inc. and Microsoft Corporation, respectively. UNIX is a trademark of AT&T Bell Laboratories.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

COMMITMENT TO PORTABILITY

STSC's proprietary APL★PLUS System spans today's entire range of computers—from desktops to the largest of mainframes.

STSC's approach to language processors provides low-cost portability from one computer architecture to another. Users of STSC's APL★PLUS Systems have the opportunity to take advantage of the most productive software—APL—and the most cost-effective hardware, simultaneously. STSC provides this flexibility through its family of systems:

- APL★PLUS Systems for the Mainframe
- APL★PLUS UNX System
- APL★PLUS PC System.

Each implementation is tailored to the architecture it runs on, taking advantage of the intrinsic capabilities of a mainframe, minicomputer, or desktop computer.

STSC's systems are implemented to be effective by themselves, but designed to interact with one another. And, micro-to-mainframe links are virtually transparent. Our built-in communications facilities give you the freedom to literally move back and forth between the micro and mainframe with a single keystroke. For example, a PC can be used as a smart work-station or terminal, interacting with a database running on an IBM mainframe computer and routing reports to a locally managed VAX.

You can link to micros and mainframes in a variety of ways. The PC can be used as a 'front end' to the mainframe, collecting data and issuing mainframe commands, such as updating a mainframe database, or making extracts from a mainframe database for analysis on the PC. Additionally, the PC can be programmed to run small jobs locally with large jobs running on the mainframe. And, all of the micro/mainframe interactions can be handled automatically so that they are invisible to the application user. The APL★PLUS PC and APL★PLUS UNX systems include state-of-the-art tools so you can upload and download files with ease. And, we provide a package to facilitate fullscreen communications at both 4800 and 9600 baud.

As hardware costs decrease and manpower costs increase, you need the flexibility to use the most cost-effective hardware that meets your needs—without rewriting your software. STSC's APL★PLUS System software provides the capabilities necessary to manage data effectively across a broad range of environments.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

THE APL★PLUS® SYSTEM UNDER MVS

STSC's APL★PLUS System under MVS is comprised of STSC's enhancements to VS APL and the APL★PLUS SHAREFILE™ System implemented in the TSO environment.

HARDWARE REQUIREMENTS

A mainframe capable of running IBM's MVS, for example:

- IBM 43XX
- IBM 370/145 or larger
- IBM 303X
- IBM 308X
- IBM 309X

Plug compatible mainframes of similar performance to the above IBM machines.

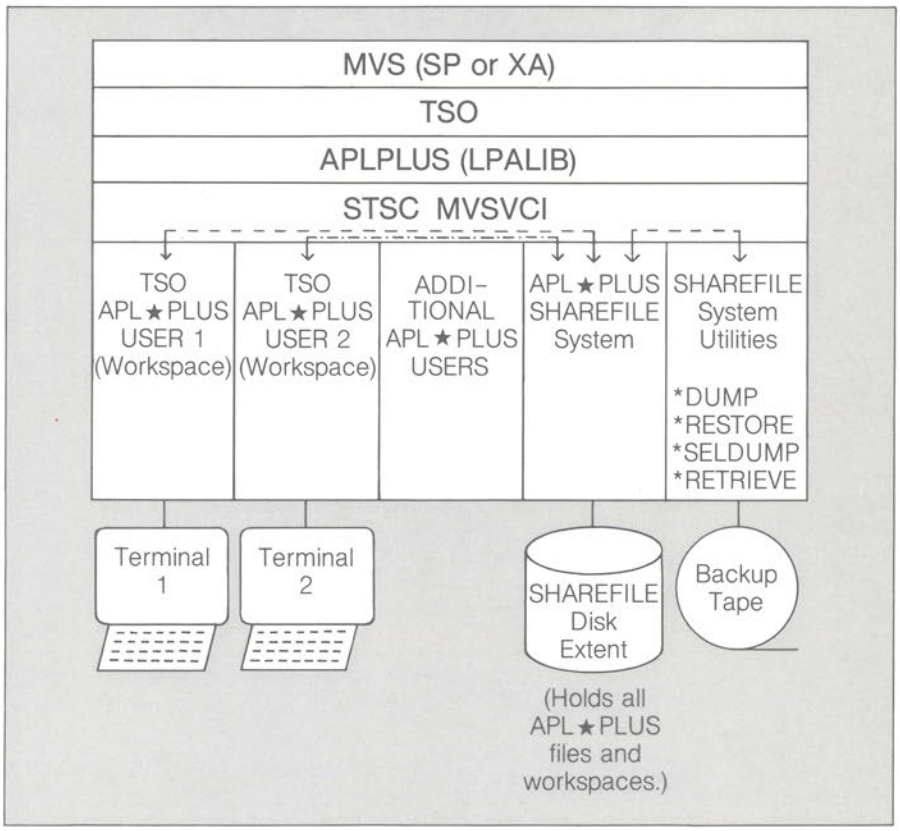
SOFTWARE REQUIREMENTS

The following IBM Program Products:

- MVS/SP Rel. 1.3 or later
- VS APL Rel. 4 (5748-AP1)
- VS APL Extended Editor IUP (5796-PLY)
- PL/1 Transient Library Rel. 4 (5734-LM5)
- GDDM (5748-XXH)—if full screen 327X support is desired.

SYSTEM COMPONENTS

The APL★PLUS System under MVS consists of several separate functional components. The first is STSC's Enhancements to IBM's VS APL program product. The second is the APL★PLUS SHAREFILE System. The shared file processor runs as a standard batch job or started task.



The diagram depicts the relationship between the APL★PLUS System and the host IBM software. No modifications to MVS or TSO code are required; a subsystem name and two type 2 SVC routines are added to MVS for inter-address space communication. Other components included are a set of SHAREFILE System utilities for backup, a set of system administrator tools (Workspace OPFNS), and a set of subroutines that provide SHAREFILE access for non-APL batch programs (FILEFNS).

APL★PLUS and IBM are registered trademarks of STSC, Inc. and International Business Machines, respectively. SHAREFILE is a trademark of STSC, Inc.
Copyright© 1985, STSC, Inc.
Printed in the U.S.A.
MA109-1985

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

THE APL★PLUS® SYSTEM UNDER VM

The APL★PLUS System under VM is comprised of STSC's Enhancements to VS APL and the APL★PLUS SHAREFILE™ System implemented in the VM/SP environment.

HARDWARE REQUIREMENTS

A mainframe capable of running IBM's VM/SP, for example:

- IBM 4331 or 4341
- IBM 370/145 or larger
- IBM 303X
- IBM 308X
- IBM 309X

Plug-compatible mainframes of similar performance to the above IBM machines.

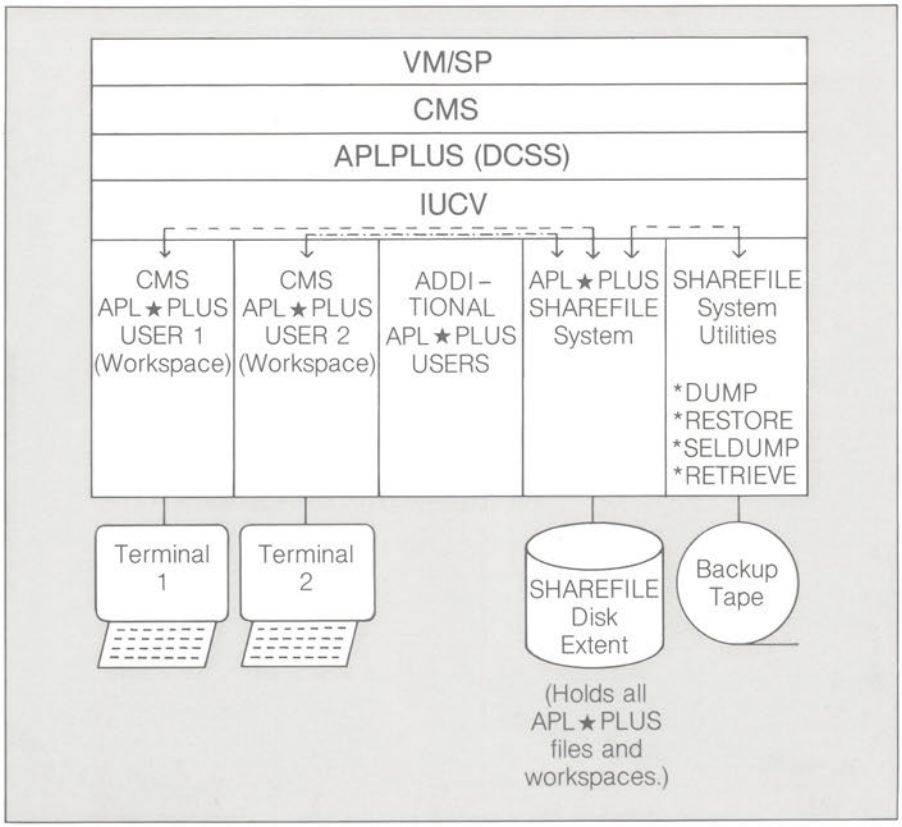
SOFTWARE REQUIREMENTS

The following IBM Program Products:

- VM/SP Rel. 3 or later
- VS APL, Rel. 4
- VS APL Extended Editor IUP (5796-PLY)
- PL/1 Transient Library Rel. 4

SYSTEM COMPONENTS

The APL★PLUS System under VM consists of several separate functional components. The first is STSC's Enhancements to IBM's VS APL program product. The second is the APL★PLUS SHAREFILE System. The shared file processor runs as a disconnected virtual machine. The diagram depicts the relationship between the APL★PLUS System and the host IBM software. No changes are required to VM or CMS. The system relies on IUCV for inter-virtual machine communication. Other components included are a set of SHAREFILE System utilities for backup, a set of system administrator tools (Workspace OPFNS), and a set of subroutines that provide SHAREFILE access for non-APL batch programs (FILEFNS).



APL★PLUS and IBM are registered trademarks of STSC, Inc. and International Business Machines, respectively. SHAREFILE is a trademark of STSC, Inc.
Copyright© 1985, STSC, Inc.
Printed in the U.S.A.
MA108-1085

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

APL★PLUS® FILE UTILITIES

STSC provides convenient and easy-to-use file system utilities for use in conjunction with the SHAREFILE™ System and other programmer tools that use files. The three utilities FILESORT, FILECONVERT, and FILEPRINT, provide an efficient means for sorting large volumes of data, converting data from an external medium to APL, and printing information stored in APL files on a high-speed printer.

FILECONVERT

FILECONVERT (formerly licensed as 'FCGEN'), is a file conversion tool which provides a means of converting non-APL datasets (comprised of records and fields) into APL files (comprised of components) that can be accessed easily by APL programs.

To accomodate this conversion, four major areas are considered:

- **Selection of Data.** You select records in the dataset to be read or skipped. You can further select the fields in the record that should be converted to APL.
- **Conversion of Data.** Each field of the record may hold character-valued data or numeric-valued data in a variety of decimal and binary representations, and you select how the file is to be converted to standard APL data representations. In addition, you can control what APL character is used for characters with no direct correspondence.
- **Rearrangement of Data.** Data will be stored in file components as numeric or character matrices, with values from successive records occupying successive rows of a matrix. You can select the order and number of components in which fields will be distributed.
- **Reporting of Erroneous Data.** Error reports are generated of incorrect characters, nonnumeric values in numeric fields, and magnetic tape errors. You select where error reports are recorded.

FILECONVERT provides a simple and uniform framework for describing the essentials of these four areas. While the FILECONVERT description is written as a short APL program, the notation used closely resembles the COBOL notation commonly used for describing record formats. The APL program is then used to automatically generate a COBOL program which is compiled and used to perform the conversion process.

The VM/CMS environment is especially well suited for its use, as FILECONVERT uses many CMS features, giving you control of the generation and execution of FILECONVERT programs. You can generate the COBOL code and then compile a FILECONVERT program.

Input can be any data that can be defined with the CMS FILEDEF command, and includes data residing in a CMS disk file, on tape, or from your virtual card reader or console.

FILEPRINT

FILEPRINT is a useful adjunct to the SHAREFILE System, allowing you to specify files from the APL environment for output at a high-speed printer. File printing is accomplished through the Auxiliary Processor AP540, which writes data from the specified file to the user's virtual printer or to the Remote Spooling Communications System (RSCS).

Workspace 1 FILEPRINT, provided with the software, contains functions for controlling the format of the printed file (handling pagination, headers, trailers, and page numbering) as well as submitting fileprint requests, and selecting appropriate translate tables for fileprint files.

The file printing facility can be modified to support the routing of files, physical devices, and print forms used at your site.

FILESORT

FILESORT provides an efficient means of sorting large volumes of data contained in APL★PLUS System files. FILESORT is a valuable tool for programmers developing applications for accounting purposes and financial and statistical analyses.

FILESORT contains two main components:

- A set of APL workspaces allowing you to set up, as well as submit, a FILESORT request.
- A batch program, which decodes the FILESORT request, handles all reference to APL★PLUS System files, and invokes the system sort utility to perform the actual sorting of the data. Any sort utility compatible with IBM's OS SORT/MERGE may be used.

COMPREHENSIVE DOCUMENTATION

STSC's APL★PLUS File System Utilities are documented with the APL★PLUS SHAREFILE System Guide (STSC, 1985), provided with the software.

AVAILABILITY

The APL★PLUS File System Utilities may be licensed as a software product for use on your own computer in the VM environment. Contact your APL★PLUS Mainframe Software Representative at (301) 984-5000 for further information.

APL★PLUS is a registered trademark of STSC, Inc.
SHAREFILE is a trademark of STSC, Inc.
Copyright© 1985, STSC, Inc.
Printed in the U.S.A.
MA107-1085

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

APL★PLUS® FILE MANAGER

STSC's APL★PLUS File Manager (formerly licensed as 'EMMA') enhances the APL programmer's ability to enter, store, manipulate, and selectively access all manner of data from APL. It extends the concept of APL primitives to database management.

The File Manager runs under the APL★PLUS mainframe environment or under IBM's APL2 product.

With the File Manager there is no need to design a file structure and access mechanism for each new application. You'll find that you can simplify the design and documentation of file structures, the programming and testing process, and the implementation of changes.

The File Manager structure is simple, and it is flexible enough to handle a wide variety of applications. It can stand alone, be used in conjunction with other files, or with other file structures. And, there is no need to understand the details of the file structure to use it.

The File Manager offers you:

- **Reliability and Stability**—The File Manager has been used continuously in a wide variety of applications since its development in 1972. And, because the file structure has remained constant, even early systems based on this technology benefit from our continued improvements.
- **Flexibility and Adaptability**—The File Manager can efficiently handle applications involving from one to over one million records. The organization of the data can be modified as changes in access needs occur. As a tool, the File Manager is as adaptable to the needs of users as APL is itself.
- **Cost-Effectiveness**—The File Manager is designed to reduce the time and cost needed to implement a new application. Compared to conventional APL systems, it frequently minimizes operating costs as well.

The File Manager contains a set of APL functions that manipulate data in the SHAREFILE™ System in a manner similar to the way in which data is handled within the workspace. Since the capabilities of these functions are based on familiar APL concepts, they are easy for the APL programmer to learn and use. The data file is logically organized as a table or matrix, and the functions that manipulate that matrix emulate APL primitives.

In the following example, a field is searched for all values meeting a certain criterion; in this case, "greater than." The resultant Boolean vector is used to select the rows (records), and the column indices (field numbers) are used to select the columns (fields). In this example, we refer to our data named 'TABLE'.

Here, assume that *FPS* represents the file parameters of the file containing *TABLE*. The statements:

```
bitvec ← (FPS fldno) GT value
data ← bitvec COMPRESS FPS. fldnos
```

correspond directly with their APL counterparts:

```
bitvec ← TABLE[;fldno] > value
data ← bitvec ≠ TABLE[;fldnos]
```

Since its introduction in 1972, STSC's File Manager has grown considerably in both scope and application. The File Manager is currently used in a variety of systems, some with storage of up to 700 million bytes. Single files can accommodate over 1 million records and over 1,000 fields.

COMPREHENSIVE DOCUMENTATION

STSC's APL★PLUS System software is delivered with complete user documentation. The File Manager documentation includes a User's Guide, Reference Manual, and a Data Selection and Formatting Guide.

AVAILABILITY

The APL★PLUS File Manager may be licensed as a software product for use on your own computer, in the VM or MVS environments. Contact your APL★PLUS Mainframe Software Representative for further information.

APL★PLUS is a registered trademark and SHAREFILE is a trademark of STSC, Inc.
Copyright © 1986, STSC, Inc.
Printed in the U.S.A.
CC243-0989

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
800-592-0050
301-984-5123 In Maryland
301-984-5412 Outside the U.S.A.

APL★PLUS® TOOLS AND PROGRAMMER UTILITIES

STSC's APL★PLUS Tools and Programmer Utility Workspaces contain many of the same features that have been used to develop and implement countless large scale applications on our APL★PLUS System.

Our software development tools and utilities are optimized, standardized, fully tested, and completely documented. They minimize the time and resources you'll spend on developing, operating, maintaining, and modifying applications. STSC's tools and utilities ensure the quick delivery of a working system.

APL Software Development Tools display and analyze components of a system, format the elements and interrelationships of the system, and edit APL functions. With these tools, you'll spend less time writing, testing, and debugging applications.

Programmer Utility Workspaces accomplish routine tasks like entering, formatting, and manipulating data. These machine-optimized utilities save you significant time and effort because they are pre-written, tested, and documented.

The APL Software Development Tools include:

- **WSDOC**—provides an automatic method of workspace documentation for program development and maintenance. WSDOC functions list the contents of a workspace in alphabetical order, by names and definitions of functions, variables, and groups. WSDOC also provides an indexed listing for each function, variable, and symbol. Output may be directed to a file for off-line printing. A version of WSDOC is also available for APL2.
- **WSSEARCH**—contains programs to aid in the debugging and editing of APL functions. Included are programs that search all specified functions for substrings or syntactic elements. Occurrences may be indicated either by listing or location, and replacements may be made automatically.
- **FNED**—edits APL functions using convenient editing commands. Programs in this workspace permit both string and syntactic element searches and replacements, single or multiple replacements, and large deletions, moves, copies, and insertions. FNED can be applied to a function in the workspace or to the character representation of a function stored as a component in the STSC SHAREFILE™ System. FNED is especially useful for extensive editing of large APL functions using dial-up teletype terminals.
- **TOOLS**—contains aids to find and reorganize names (identifiers) used in APL functions. TOOLS makes it easy to keep functions "clean" as they are being developed and maintained by taking the tedium out of common editing tasks, such as the reordering of labels.

The Programmer Utility Workspaces include:

- **DATES**—contains functions to convert, format, and verify dates and to perform miscellaneous manipulations on dates based on the Gregorian calendar. For example, functions are included to:
  - — convert numeric dates and timestamps to various formats
  - — calculate differences between dates
  - — compute the day of the week for a current date
  - — verify dates, leap years, etc.
- **INPUT**—contains functions that collect and process input that can be incorporated into APL programs. The input functions in this workspace display programmer-defined prompts and accept character input, bulk input, and combined character, numeric, and forced-choice input.
- **SORT**—contains utilities for grading, ranking, and sorting of both character and numeric arrays. One group of functions handles numeric arrays as a collection of independent row vectors; and a second group is used to grade, rank, and sort rows of a matrix. Designed primarily for matrices, the functions are implemented to conserve storage and operate efficiently on large arrays.
- **PARTFNS**—contains user-defined functions that apply certain APL primitive functions independently to defined partitions of an array. These functions are used to manipulate a collection of arrays as independent objects, although the collection is stored in a single array.

COMPREHENSIVE DOCUMENTATION

STSC's APL★PLUS Tools and Programmer Utility Workspaces are completely documented online and in the APL★PLUS System Tools Manual (STSC, 1985), provided with the software.

AVAILABILITY

The APL★PLUS Tools and Programmer Utility Workspaces may be licensed as software products for use on your own computer in the VM or MVS environments. Contact your APL★PLUS Mainframe Software Representative at (301) 984-5000 for further information.

APL★PLUS is a registered service mark and trademark of STSC, Inc.
SHAREFILE is a trademark of STSC, Inc.
Copyright© 1986, STSC, Inc.
Printed in the U.S.A.
MA105-0786

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

APL ★ PLUS® COMPILER

STSC's APL★PLUS Compiler, the first APL compiler to be offered commercially, is the result of six years of pioneering research and development at STSC. It runs on the Mainframe APL★PLUS System.

The compiler is a tool for speeding up programs and reducing CPU costs. How much faster do compiled programs run? The speedup varies depending on the program and the nature of the problem. Customers report that entire applications are from 20% to 9 times faster when compiled, and that individual functions are up to 100 times faster.

USEFUL FOR A BROAD RANGE OF APPLICATIONS

Early customers have compiled a diverse range of applications including:

- financial portfolio analysis
- analysis of stock trading strategies
- inventory applications
- manufacturing applications
- text processing
- transaction parsing
- structural engineering
- electronic circuit design.

Feedback from users has been very favorable. They are pleased with the compiler's performance and say that it does what they expected it to do. Some users report that they now write programs they would not have even attempted without the compiler.

The compiler plays two important roles: not only is it a tool for speeding up existing applications, but it can also be used to write new programs that would be unacceptably slow if interpreted.

SPEEDS UP EXISTING APL PROGRAMS

The compiler speeds up programs mainly by reducing interpretive overhead, so programs with large amounts of interpretive overhead (programs involving small arrays or looping) may be sped up dramatically. Programs with little interpretive overhead, though sped up less, can still be significantly faster when compiled. For heavily used applications, even a modest speedup can represent a significant savings in CPU time.

MAKES APL PRACTICAL IN NEW AREAS

Some problems can be solved in APL without looping, but it may take an APL expert a long time to figure out how. Often, the resulting programs are incomprehensible to other programmers and difficult to modify. Other problems that are inherently iterative simply cannot be solved efficiently with an interpreter. To solve such problems without an APL compiler, you must either tolerate the inefficiency or use a different language. A compiler is practically essential for applications such as:

- transaction-oriented systems
- simulation
- first-in first-out inventory accounting
- resource scheduling
- time-series analysis
- exponential moving averages
- linear programming
- differential equations
- nonlinear equations
- finite element analysis
- linked lists
- network and graph flow problems
- compilers and assemblers.

SAVES PROGRAMMER TIME

By reducing interpretive overhead, the compiler makes it unnecessary to use a special programming style adapted to the interpreter. This makes it possible to write a wider range of efficient programs that are clear, forthright, correct, and easy to maintain. The compiler can save programmer time in three ways: (1) it can reduce the time spent seeking more efficient solutions, (2) it can save maintenance time by making straightforward solutions practical, and (3) it can allow you to use APL in places where you might otherwise have to use a more cumbersome language.

COMPILED FUNCTIONS

Compiled functions can use any APL operations included in the proposed ANSI/ISO APL standard. To the user, compiled functions appear very much like interpreted functions. Their names show up in the)*FNS* display, they can be saved in workspaces, copied from workspace to workspace, or moved to and from files. The source code of the original program is imbedded in the compiled function, so a separate source copy need not be maintained. When a compiled function encounters an error, it produces an ordinary APL error message, not a system error.

HOW THE COMPILER IS USED

Because the compiler is used in conjunction with the interpreter, it retains the benefits of the interactive APL en-

vironment for application development. Applications are written and debugged as usual with the interpreter. When the application is complete, the monitoring facility (*MF,* which is included with the APL★PLUS Enhancements System) is used to find the programs that consume the most time (the "CPU bottlenecks"). Usually, only a few functions in an application need to be compiled to achieve most of the speedup possible. Compiling just the CPU bottlenecks saves time and makes the application easier to debug and change.

FASTFNS

The APL★PLUS Enhancements System includes a collection of precompiled utility functions (FASTFNS). Each function in the FASTFNS workspace uses the most efficient algorithm known and is significantly faster than a similar interpreted APL program. Dozens of functions are provided that perform common and useful operations such as:

- join matrices top-to-bottom, padding with blanks as required
- convert a list of names to a matrix
- matrix row lookup
- delete excess blanks
- perform datatype conversion
- compute CRC checksums.

These compiled functions can be imbedded in applications to provide immediate speedups at minimal cost.

COMMONLY-ASKED QUESTIONS

- **Is the Compiler Hard to Use?** No! Our customers have reported several cases where applications that took two or three years to develop were speeded up in only a few days using the compiler.
- **Are Declarations Required?** Some declarations are required (for external objects) but not as many as are required for most conventional languages. You provide enough declarations to give the compiler a start, and it figures out the rest.
- **How Much Time Does It Take to Compile a Function?** It varies with the size and complexity of the program and with the speed of your computer. However, when the compiler is applied to programs that use a large amount of CPU time, the cost of compilation can be paid back with just a few runs of the compiled APL program. Also, the Request System provided with the compiler may allow you to utilize unused CPU time late at night, effectively compiling functions for free.
- **What Do I Need to Be Able to Use the Compiler?** You need the APL★PLUS Enhancements System, the SHAREFILE™ System, and the APL★PLUS Compiler.

COMPREHENSIVE DOCUMENTATION

The *APL★PLUS Compiler Reference Manual* (STSC, 1985) is a comprehensive guide to using the compiler. It tells you how to choose which functions to compile, how to prepare them for compilation, how to optimize them to get the most out of the compiler, and what error messages mean. Another document, the *APL★PLUS Compiler Installation Guide* (STSC, 1985), provides step-by-step instructions for installing and maintaining the compiler.

AVAILABILITY

The APL★PLUS Compiler is available as a licensed software product for use on your own computer in the VM or MVS environments. Contact your APL★PLUS Mainframe Software Representative at (301) 984-5000 for further information.

APL★PLUS is a registered service mark and trademark of STSC, Inc.
SHAREFILE is a trademark of STSC, Inc.
Copyright©1986, STSC, Inc.
Printed in the U.S.A.
MA110-0786

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

APL★PLUS® ENHANCEMENTS

STSC's premier mainframe software product is a collection of proprietary enhancements to IBM®'s VS APL program product. The Enhancements Product has evolved over many years as a result of the needs of thousands of users of the APL language. Each enhancement provides a feature of genuine value toward improving programmer productivity, extending the ability to create more sophisticated APL applications, or improving machine efficiency.

The components of the Enhancements Product can be described as follows:

- **Language Extensions**—Improvements to many of the features found in VS APL, extensions to the capabilities of certain APL operations, and the addition of many new language features.
- **Nested Arrays**—A major extension to the language that allows data structures far more flexible than those allowed by traditional APL. New APL primitive functions have been created to perform operations on nested structures.
- **Report Formatting**—The $\square FMT$ system function, which lets you produce presentation quality business reports by formatting both character and numeric data at one time. Combines the best of FORTRAN formatting notation and COBOL picture editing with APL array handling. A special version of $\square FMT$ is available for IBM's APL2 product.
- **Exception Handling**—Features that automate and expand the system's ability to react to errors and exceptions that can occur during program execution. When used by APL programs, these features increase reliability, security, and efficiency by allowing programs to react to errors in a controlled fashion.
- **FASTFNS**—A collection of pre-compiled utility functions that run much faster than equivalent interpreted functions. Provided in response to user suggestions, these pre-compiled programs perform various useful operations like date manipulation and character text searching.
- **Monitoring Facility**—The $\square MF$ system function, which provides an easy way to find where CPU time is spent in an application so you can concentrate speedup efforts on those areas that will make the most difference.
- **System Functions and Variables**—A variety of new system functions and variables for processing character data, managing user-defined functions, and controlling the workspace and system environment.

The Enhancements Product is described in more detail in the following sections.

APL LANGUAGE EXTENSIONS

The Enhancements Product includes many system features that make it easy to write and maintain APL programs.

- The diamond statement separator (◇) lets you form compound statements on a function line. Facilitates writing short, easy-to-read expressions, making programs easier to understand and maintain.
- Trailing comments following APL statements on a function line make it more convenient to document the intent of each line.
- Text insertion with period (.), comma (,), or semicolon (;) makes line editing quicker.
- Domain and value checks on assignment for system variables allow you to identify problems immediately upon assignment.
- Replication, a consistent extension to compression, extends the domain of the left argument to the slash operation (/) to nonnegative integers, allowing compression to make an arbitrary number of copies of each element.
- Enhancements to upgrade and downgrade extend them to character data and arrays of rank greater than one, allowing rows of a table to be sorted.
- Dyadic user-defined functions can be called either dyadically or monadically. These "ambivalent functions" allow you to write programs that have a default left argument that can be overridden by the user in special cases.
- Extensions to the character set to

include the dollar sign, cents sign, and a variety of international symbols for clearer, more accurate reports.

NESTED ARRAYS

Nested Arrays allow data structures far more flexible than those allowed by traditional APL. Traditional APL allows you to create, for example, a two-dimensional array (a matrix or table) which has as each of its items a single number or single character. With nested arrays, each item of an array can itself be an array. Thus, arrays may be nested within other arrays.

Some of the specific benefits of this new data structure are

- You can now mix character and numeric data since nested objects can contain more than one datatype.
- You can effectively pass a larger variety of data to a function by forming the arguments into a nested vector.
- Nested structures allow APL to match the features of other languages since they provide a recursive data structure more powerful than LISP and allow you to mimic the data structures of languages like COBOL.

Nested arrays features incorporate new APL primitive functions, new data types, and a new syntax which broadly expand the power of the language and the ways in which an APL programmer may approach a problem. It is generally considered the most significant extension to APL since the language's inception—a tool which adds new dimensions to the productivity already possible with APL.

REPORT FORMATTING

The system function ⎕*FMT* is a powerful, general purpose, business report formatting facility that combines the best of FORTRAN formatting notation and COBOL pictured editing. ⎕*FMT* gives you the ability to:

- Include titles and column names in a finished report.
- Combine character text with numeric values, such as a name and telephone number side by side.
- Display data in a more recognizable form, such as financial data with dollar and cent signs and commas.

In an APL environment, character and numeric data are handled differently on display. Character data is displayed subject only to your terminal width setting and conventions for displaying matrices and higher dimensional arrays. Numeric data is formatted according to conventions adopted by the APL★PLUS System. ⎕*FMT* is a formatting tool that converts data to a character image according to the specifications you provide. These specifications control the spacing, representation, precision, and decoration of the character image.

A one-line ⎕*FMT* formatting statement can turn a table of meaningless numbers into a timely report. For example, the variable *NUM* below is turned into a table that uses dollar signs, places negative values in parentheses, and specifies a field width of 20 with two decimal places:

```
    NUM
1316026          755715.3407   4587093.116
  11586.9       2190044.457    ¯471009.5613
6789137.817            0       9347189.774
3836077.871     ¯5195105.458        345.5

      'C P¨$¨ Q¨ ¨ M¨($¨ N¨)¨ F20.2' ⎕FMT NUM
$1,316,026.72          $755,715.34        $4,587,093.12
   $11,586.90        $2,190,044.46         ($471,009.56)
$6,789,137.82                $0.00        $9,347,189.74
$3,836,077.87       ($5,195,105.46)              $234.50
```

COMPILER TECHNOLOGY

STSC has introduced in the Enhancements Product technology that gives you the ability to use pre-compiled functions and incorporate them into applications. The benefits of compiled APL programs are evident in the increased efficiency of your applications. How do you determine where your applications are slow? You can monitor your functions to find out exactly where your applications need speeding up by using the ⎕*MF* system function. Pre-compiled functions and ⎕*MF* are explained in more detail below.

Pre-Compiled FASTFNS

You can enjoy the benefits of faster APL program execution immediately by using the precompiled functions supplied with the Enhancements Product. These functions, called FASTFNS, implement popular utility operations using the fastest known algorithms.

Most of the functions provided with the FASTFNS workspace are far faster than other APL functions that do the same job. Some are compiled APL functions, while others are written in assembly language or our own systems programming language called TABL (for Tool and Application Building Language). The functions in the FASTFNS workspace are typically three to thirteen times faster than their standard APL alternatives, although some are up to several hundred times faster.

There are presently more than three dozen functions in the FASTFNS workspace. They are helpful for checking input, formatting output, text processing, sorting, ranking, counting, table lookup, communications, and converting data from non-APL sources. New FASTFNS are added to the workspace with every release of the Enhancements Product.

Finding the CPU Bottlenecks

Knowing where CPU time is spent in an application is essential to being able to speed up the application. Because of this, we have provided the ⎕*MF* monitor function, which provides an easy way to find the CPU "bottlenecks". Finding the slowest (costliest) functions in a workspace allows you to concentrate speedup efforts on those areas that can make the most difference.

Workspace 3 MFFNS, provided with the Enhancements Product, contains utility functions which use ⎕*MF* to produce formatted reports summarizing CPU usage. Once the bottlenecks in an application have been identified, you'll know exactly which APL programs to optimize, either by using your own speedup methods, by incorporating pre-compiled FASTFNS, or by compiling the programs using STSC's APL★PLUS Compiler (described in a separate brochure).

EXCEPTION HANDLING

STSC's Exception Handling facility both automates and expands the system's ability to react to errors and exceptions that can occur during program execution. Exception Handling, or "error trapping", can be used to increase the reliability and efficiency of APL programs by allowing the program to react to errors in a controlled fashion. It can also improve the security of an application, protecting the application and the data it contains from accidental or malicious interference. In short, exception handling can be used to write applications that are tolerant of users who have minimal training on the system.

Exception Handling can also improve the efficiency of programs by avoiding the costy prechecking of each function argument or each element of data. For example, instead of checking all terminal input, an application could be written to invoke an error-checking routine only when an invalid entry causes an execution-time error.

The Exception Handling facility comprises four system variables—⎕*ELX*, ⎕*ALX*, ⎕*DM*, and ⎕*SA*—and one system function—⎕*ERROR*. Their possible applications for error handling are almost limitless.

These functions provide significantly more flexibility in handling errors than is possible with VS APL. For example, the error trapping facility provided by VS APL (⎕*EA*) cannot be used in immediate execution mode to trap errors outside of programs. The absence of ⎕*DM* in VS APL makes it impossible for a program to determine what kind of error occurred, increasing the likelihood that inappropriate action will be taken when an unexpected error occurs.

⎕*ERROR* is also an indispensable tool for error trapping and debugging.

For example, if you are stopped in a subroutine and need to climb the state indicator to a calling routine, □*ERROR* ' ' can be used to remove one level from the state indicator. In VS APL, you could erase the result of the current function and exit, but this will not work if there is no result or if the caller just prints the result. Another method in VS APL would be to put a stop set on the next line of the calling program and then exit. However, this will not work if you are on the last line of the caller, if the current line does a branch, or if the current line involves several calls to the subroutine. □*ERROR* makes this common operation easy.

Following is a summary of the Exception Handling functions:

□*ALX* Attention Latent Expression; a latent expression that is executed in the event of a weak interrupt. For example, □*ALX* can prevent the user from interrupting execution at an inappropriate place.

□*DM* Diagnostic Message; returns the last diagnostic message recorded in the workspace. Can be used by error-handling programs to diagnose the error that just occurred. Also useful when an application keeps a log of errors that occur.

□*ELX* Error Latent Expression; a latent expression that is executed in the event of certain errors. Used to specify a program's automatic response when an error occurs in the program.

□*SA* Stop Action; specifies a "last resort" action that is taken if a task enters immediate execution input mode. Provides an extra level of foolproof application security.

□*ERROR* Removes the function executing it from the execution stack and signals an error exception.

ADDITIONAL SYSTEM FUNCTIONS AND VARIABLES

STSC provides a wide variety of new system functions and variables—over and above what's available with VS APL. System functions included with the Enhancements Product are grouped below into functional areas and briefly described.

Processing Character Data and Format Conversion

The system functions described below enhance the system's ability to convert between character and numeric data and perform other useful processing on character data.

□*FI* Converts free-form character data (typically entered as input by the user) into numeric data. More secure than evaluated (□) input.

□*VI* Verifies the conversion of character strings into numeric data. Usually used in conjunction with □*FI*.

□*SS* Performs a string-search operation. Simplifies the design of many text processing programs and executes three to four times faster than the fastest known APL equivalent.

Improved Management of APL Functions

The system functions described below enhance the programmer's ability to define and edit APL functions under program control.

□*CRL* Returns the canonical representation (character vector) of a single line of a function.

□*DEFL* Defines or edits a single line of a function. Allows the dynamic modification, insertion, or deletion of lines in a function by another running function.

□*VR* Returns the visual representation (character vector) of a function. Used to store functions on file (which can then be redefined by □*DEF*). More

convenient than □*CR* because function representation includes line numbers.

□*CRLPC* Returns the canonical representation of a public comment on a line of a (possibly locked) function. A public comment is one which begins with the lamp and del (⍝∇) symbols.

□*LOCK* Locks the definitions of a list of functions.

□*DEF* Defines a function from either the canonical or the visual representation of the function.

Terminal Control Characters

The following "read only" system variables can be used as character constants. These characters are used to control the position and movement of the cursor or print mechanism during output to the terminal, or to communicate with a terminal or other intelligent device through "escape sequences".

□*TCBS* Contains a backspace character.

□*TCDEL* Contains a delete character.

□*TCLF* Contains a linefeed character.

□*TCNL* Contains a new-line character.

□*TCNUL* Contains a null character.

Workspace and System Environment

The system functions, commands (denoted by a right parenthesis), and variables described below control the workspace environment or inquire about the workspace and system environment.

□*ARBOUT* ⍳0 Clears the quote-quad input buffer.

□*ERASE* Erases, if possible, the most local definitions of a list of identifiers in the active workspace.

□*GRP* Returns the canonical representation (character matrix) of a group definition. Used to find the definition of a group under program control.

□*IDLIST* Returns a list of names for all objects of a given class or classes in the active workspace.

□*IDLOC* Returns the local and global classifications for each name in a list of names.

□*LOAD* Replaces the active workspace by a designated workspace. Allows an application to switch to a new workspace automatically.

□*QLOAD* Same as □*LOAD* except the SAVED message is suppressed (i.e., the workspace is quietly loaded). □*QLOAD* can be used to conceal a change of workspace from the user.

□*XLOAD* Same as □*LOAD*, but suppresses execution of the latent expression □*LX* when the new workspace is loaded. Also provided is the system command *)XLOAD*.

□*SIZE* Returns the amount of space used by a list of identifiers. Useful in programs that report on the contents of a workspace.

)EDIT Displays any of the last five lines of immediate execution input and makes those lines available for editing or re-execution.

)RESET Provides a convenient way to clear the state indicator in the active workspace.

)STORE Allows you to store objects in a workspace other than your active workspace.

The following "read only" system variables provide useful information under program control or in immediate execution mode:

- *WSSIZE* Returns the size of the active workspace. Used by applications that keep track of or control the active workspace size.
- *WSTS* Returns the time at which the workspace in use was last saved. Can be used by a running program to identify the version of the application being used.
- *SYMB* Returns the current number of symbol table entries, allowing a running program to keep track of symbol table size. Also can be used by programs that report on the contents of a workspace.
- *SYSID* Returns the identification of the APL★PLUS System in use.
- *USERID* Returns the account name of the user.
- *WSID* Returns the identification of the active workspace. Used to determine the workspace ID under program control.
- *CPUID* Returns numeric identification of the central processing unit (CPU) in use.
- *SI* Returns the contents of the state indicator: a list of suspended and pendent functions. Enables a running program to examine and analyze the execution stack.
- *SSL* Irreversibly super seals a workspace so you can hide names within an application from users or prevent users from copying objects from the workspace.

PC/MAINFRAME COMPATIBILITY

Most of the features of the Enhancements Product are available on the APL★PLUS PC System for microcomputers and the APL★PLUS UNX System for supermicros and minis. Using PC's or supermicros as workstations can improve programmer productivity and help reduce mainframe load. Programs can be prototyped on smaller machines, then uploaded to the mainframe to run because you have practically identical APL systems on both machines.

COMPREHENSIVE DOCUMENTATION

Documentation for each of the features described here is provided in **APL★PLUS Enhancements** (STSC, 1985).

AVAILABILITY

The Enhancements Product may be licensed as a software product for use on your own computer in the VM or MVS environments. Contact your APL★PLUS Mainframe Software Representative at (301) 984-5000 for further information.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

APL★PLUS is a registered service mark and trademark of STSC, Inc. IBM is a registered trademark of International Business Machines.
Copyright © 1986 STSC, Inc.
Printed in the U.S.A.
MA103-0786

A PLUS★WARE™ PRODUCT

SHAREFILE™—THE APL★PLUS® SHARED FILE SYSTEM

With SHAREFILE, the APL★PLUS Shared File System, STSC provides a secure, shared file system to the VS APL and APL2 environments. SHAREFILE provides exceptional flexibility and control in sharing data and applications among users.

This component-oriented file system extends the capabilities of APL so that a user can enter, retain, and access more data than can be stored in an APL workspace; share the database with other users and allow selective access and modification.

In addition, SHAREFILE is able to interface with user programs written in other languages—non-APL programs can acccess the same files as APL programs.

In the APL2 environment, SHAREFILE uses standard IBM interfaces to provide APL data manipulation and storage features, without modifications to APL2. SHAREFILE is designed to take advantage of many of the inherent properties of APL2. For example,

- SHAREFILE will handle any APL2 data object, including general (nested) arrays, complex numbers, and extended character features.
- SHAREFILE takes advantage of APL2's event signalling and reporting mechanism. Error messages and completion codes fit into the APL2 error message scheme.

EASE OF USE

SHAREFILE allows you to handle large amounts of data without knowledge of the underlying disk storage devices. Storing and retrieving APL data items on auxiliary storage outside the workspace is achieved easily. There is no need to translate an item into a particular data type or to block (or segment) the data item into a fixed length.

Each data item is written as a component of a file and each component may hold any array, regardless of what is in other components. Components can be replaced by other items with no length or data-type restrictions imposed by the previous value of the component.

The ease of using SHAREFILE allows programmers in interactive environments to focus on the application, rather than on operating system restrictions on using files.

The following sample terminal session shows all the commands necessary to read data from a file created by another user, extract the desired fields, and write the data to a new file just created. In the APL2 environment, you are provided with a workspace of the same easy-to-use file commands as shown below, except the "quad" (⎕) symbol is not used in the command name.

'1 2 3 4 OLDFILE' ⎕FSTIE 1

Activate, or "tie" file *1 2 3 4 OLDFILE* using tie number 1. This tie number is entirely arbitrary.

X←⎕ *FREAD 1 28*
Read data from component 28 of the file tied to 1. The owner of file 1 must have authorized access prior to the read.

Y←*X (3; 1 2 4 7 14)*
Select desired fields.

'NEWFILE' ⎕ *FCREATE 10*
Create a new file called *NEWFILE* and reference it as file 10.

Y ⎕ *FAPPEND 10*
Append *Y* to file 10.

You need not be concerned with what device holds the data, or even whether the data is character, integer, or floating point. You only need to know which component contains the data and which fields are needed.

Similar access is available to COBOL, PL/1, and FORTRAN programs using subroutines included with SHAREFILE.

FILE SHARING

SHAREFILE allows full file-sharing capabilities, a feature not generally available in IBM's VM or MVS environments.

No significant limitation exists to the number of users that can access a file at the same time. Easy-to-use control tools allow complete read and write sharing of files. Once you designate certain files as shared files, there is no special action required when others use the files. And, in addition to sharing your files, you can designate those files which are not to be accessed.

SECURITY

Even with its ease-of-use and sharing capabilities, SHAREFILE maintains ownership control over all files, permitting authorized access only.

File security is accomplished using a mechanism called an access table which is initially empty. Upon creation, a file can be accessed only by its owner. If the owner wishes to allow another user to access that file, an entry must be made in the access table specifying the other user's identification and what type or types of access will be permitted. Numerous levels of access, such as read, write, and erase, can be granted in any combination.

SHAREFILE extends access table control to APL workspaces as well. APL workspaces are saved as SHAREFILE files and enjoy the same flexibility and security of access.

ACCOUNTABILITY

The SHAREFILE System provides you with file storage and file resource unit accounting information, giving you the opportunity to monitor SHAREFILE System usage at your site, and alternatively, "chargeback" your users for resources consumed.

File storage accounting information reports the total bytes owned by a user. This information is automatically recorded every time the SHAREFILE System is started. File Resource Units (FRUs) are calculated on each file operation performed by a user and represent work done by the SHAREFILE System Server Machine in processing a user's work.

MINIMUM OPERATOR INTERVENTION

The SHAREFILE System is designed to run with a minimum of operator intervention. The system runs as a disconnected virtual machine or as a background processor. The only operational tasks requiring attention are STARTUP and SHUTDOWN, and running utility programs. A complete set of system utilities is provided

with SHAREFILE for such tasks as BACKUP and RESTORE.

RELIABILITY, MAINTAINABILITY, EFFICIENCY

While SHAREFILE is written in PL360 for reliability and maintainability, it is also very efficient. Certain key modules are optimized in 370 Assembly language to ensure superior performance. By design, SHAREFILE is extendable to new I/O devices in a much easier fashion than was possible in earlier APL file systems.

COMPATIBILITY

STSC's systems software and applications software are based on over 15 years of development efforts in its core technology—APL. Over the years STSC has improved the performance and extended the range of APL with proprietary systems software packages such as SHAREFILE. Our development has enabled us to offer information processing systems that are general enough to support a wide range of users, yet can be customized more easily than traditional packaged software. SHAREFILE is designed to be compatible with all of STSC's software.

Since STSC introduced the first shared access, component-based APL file system in 1969, it has become the industry standard for state-of-the-art file system design. Over the years, the SHAREFILE System has been adapted to various machine architectures and environments. SHAREFILE is available with STSC's APL★PLUS System running on micro, supermicro, mini, and mainframe computers in the MS-DOS®, UNIX™, VM/CMS, and MVS/TSO environments.

SHAREFILE COMMANDS

Function	Purpose
☐ *ACODE*	Converts operation names to numeric access codes.
☐ *FAPPEND*	Adds a component to the end of a file.
☐ *FAVAIL*	Determines if the SHAREFILE System is available.
☐ *FCREATE*	Creates a new file.
☐ *FDROP*	Drops components from the beginning or end of a file.
☐ *FERASE*	Erases a file.
☐ *FHOLD*	Holds a file for updating.
☐ *FLIB*	Lists files in a library.
☐ *FNAMES*	Lists names of tied files.
☐ *FNUMS*	Lists tie numbers of tied files.
☐ *FRDAC*	Reads the access table of a file.
☐ *FRDCI*	Reads information about a component of a file.
☐ *FREAD*	Reads a component from a file.
☐ *FRENAME*	Renames a file.
☐ *FREPLACE*	Replaces a value into an existing component of a file.
☐ *FRESIZE*	Changes the size limit of a file.
☐ *FSIZE*	Queries the size of a file.
☐ *FSTAC*	Sets the access table of a file.
☐ *FSTATUS*	Determines status of a particular file (for example, who has the file tied, or who has a file held).
☐ *FSTIE*	Ties, or activates, a file for shared use.
☐ *FSWITCH*	Selects among several SHAREFILE Systems which may be concurrently executing.

□ *FTIE*	Exclusively ties, or activates, a file for subsequent use.
□ *FUNTIE*	Unties one or more files from the active list.
□ *STORAGE*	Displays total amount of storage used by files and APL workspaces and the current storage limit

STSC has enhanced standard APL system functions to access workspaces and libraries stored under SHAREFILE control. Workspace-related system functions and variables available with SHAREFILE are described briefly below.

□ *LRDAC*	Returns the access matrix for a library.
□ *LSTAC*	Sets the access matrix for a library.
□ *WSOWNER*	Returns the user number of the owner of the active workspace.
□ *WSRDAC*	Reads the access table of stored workspace.
□ *WSRENAME*	Renames a saved workspace.
□ *WSSTAC*	Sets the access table of a stored workspace.
□ *LOAD*	Loads a new workspace under program control.
□ *QLOAD*	Loads a new workspace without printing the usual message.
□ *XLOAD*	Loads a new workspace without executing its latent expression.
□ *WSLIB*	Returns a list of workspaces currently stored.

COMPREHENSIVE DOCUMENTATION

STSC's SHAREFILE System is extensively documented in the APL★PLUS SHAREFILE System documentation (STSC, 1985), provided with the software.

AVAILABILITY

The APL★PLUS SHAREFILE System may be licensed as a software product for use on your own computer. Contact your APL★PLUS Mainframe Software Representative at (301) 984-5000 for further information.

APL★PLUS is a registered trademark of STSC, Inc.
SHAREFILE is a trademark of STSC, Inc.
UNIX is a trademark of AT&T Bell Laboratories.
MS-DOS is a registered trademark of Microsoft Coporation
Copyright©1985 STSC, Inc.
Printed in the U.S.A.
MA104-1085

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

A PLUS★WARE™ PRODUCT

STSC
A CONTEL COMPANY

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852

A PLUS★WARE™ PRODUCT

APL★PLUS is a registered trademark of STSC, Inc.
SHAREFILE and PLUS★WARE are trademarks of STSC, Inc.

Copyright 1985
STSC, Inc. Printed in the U.S.A.

MA101-0585

Are You Ready For APL2 ?

STSC
A CONTEL COMPANY

STSC INTRODUCES SHAREFILE™/AP SHARED FILES FOR THE APL2 ENVIRONMENT

STSC has introduced a new secure, shared access, component-based APL file system for the APL2 environment. SHAREFILE/AP uses standard IBM interfaces to provide APL data manipulation and storage features, without modifications to APL2.

SHAREFILE/AP combines many of the superior capabilities of STSC's standard SHAREFILE System with a sophisticated auxiliary processor design, providing the features and benefits you have come to expect from a component-based APL file system.

Fully Integrated With APL2

SHAREFILE/AP is designed to take advantage of many of the inherent properties of APL2. For example,

- SHAREFILE/AP will handle any APL2 data object, including general (nested) arrays, complex numbers, and extended character features. Data is not translated or truncated in any way.
- SHAREFILE/AP takes full advantage of APL2's event signalling and reporting mechanism. Error messages and completion codes fit right into the APL2 error message scheme.
- SHAREFILE/AP supports international language translations, which can provide you with error messages and command names in languages other than English.

Straightforward Design, Easy To Use

A workspace of easy-to-use file functions is provided with SHAREFILE/AP. This workspace allows you to use simple names like FREAD, FAPPEND, and FDROP to access and update your data.

Since SHAREFILE/AP uses a mnemonic protocol to perform file system operations, you can access the shared variable directly using operation names rather than numeric codes. The shared variable protocol uses a single variable, single thread design, which allows fast, efficient, and safely interruptable operations because only one assignment of the shared variable is required to perform an operation. You need not worry about the possibility of spurious file damage after the resumption of an interrupted file operation.

SHAREFILE/AP — A Sound Investment

Additional features and benefits of SHAREFILE/AP that make it a valuable investment for the APL application developer can be found in STSC's SHAREFILE System for any environment:

- SHAREFILE/AP supports very large files and can handle vast quantities of data with no degradation in performance. Each individual file can contain over two gigabytes of data, and there is no practical limit on the number of files the system can contain.
- Files are easily shared with other users, even concurrently. SHAREFILE/AP permits many users to read and update the same file simultaneously without delay or damage to data.
- SHAREFILE/AP offers exceptional security and file control. The creator of a file has exclusive access by default; others can access files only if the owner grants explicit permission. The owner has a fine degree of control over use of his files, letting, for example, others append data to a file but not read or modify the contents in any other way. Additionally, programs written in other languages, such as FORTRAN, COBOL, PL/I, or Assembler, can read, write, create, erase, and otherwise access shared files.

The Test of Time

Since STSC introduced the first shared access, component-based, APL file system in 1969, it has become the industry standard for state-of-the-art file system design. Over the years, the SHAREFILE System has been adopted to various machine architectures and environments. SHAREFILE is also available with STSC's APL★PLUS® System running on micro, supermicro, mini, and mainframe computers in the MS DOS, UNIX, VM/CMS and MVS/TSO environments.

STSC, the Company. . .

STSC, A Contel Company, is the industry leader in the development of the APL programming language. STSC delivers APL-based software for microcomputer and mainframe environments, and publishes microsoftware written by independent developers. STSC has branch offices nationwide and licenses software through distributors in several countries worldwide.

STSC

APL*PLUS® Mainframe Software

SHAREFILE™/AP

Shared File System For APL2®

With SHAREFILE/AP, STSC provides a secure, shared file system to the APL2 environment. SHAREFILE/AP provides exceptional flexibility, performance, and control in sharing data and applications among users.

This component-oriented file system extends the capabilities of APL2 so that a user can enter, retain, and access more data than can be stored in an APL2 workspace, share the database with other users, and allow selective access and modification.

In addition, SHAREFILE/AP is able to interface with user programs written in other languages -- non-APL programs can access the same files as APL programs.

In the APL2 environment, SHAREFILE/AP uses standard IBM interfaces to provide APL data manipulation and storage features without modifications to APL2. SHAREFILE/AP is designed to take advantage of many of the inherent properties of APL2. For example:

SHAREFILE/AP will handle any APL2 data object, including general (nested) arrays, complex numbers, and extended character features.

SHAREFILE/AP takes advantage of APL2's event signalling and reporting mechanism. Error messages and completion codes fit into the APL2 error message scheme.

EASE OF USE

SHAREFILE/AP allows you to handle large amounts of data without knowledge of the underlying disk storage devices. Storing and retrieving APL data items on auxiliary storage outside the workspace is achieved easily. There is no need to translate an item into a particular data type or to block (or segment) the data item into a fixed length.

Each data item is written as a component of a file and each component may hold any array, regardless of what is in other components. Components can be replaced by other items with no length or data-type restrictions imposed by the previous value of the component.

The ease of using SHAREFILE/AP allows programmers in interactive environments to focus on the application, rather than on operating system restrictions for using files.

The following sample terminal session shows all the commands necessary to read data from a file created by another user, extract the desired fields, and write the data to a new file just created:

```
'1234 OLDFILE' FSTIE 1
```
Activate, or "tie" file *OLDFILE* in library 1234 using tie number 1. This tie number is entirely arbitrary.

```
X←FREAD 1 28
```
Read data from component 28 of the file tied to 1. The owner of file 1 must have authorized access prior to the read.

```
Y←X[3;1 2 4 7 14]
```
Select desired fields

```
'NEWFILE' FCREATE 10
```
Create a new file called *NEWFILE* and tie it as file 10.

```
Y FAPPEND 10
```
Append data in *Y* to file 10.

You need not be concerned with what device holds the data, or even whether the data is character, integer, or floating point. You only need to know which component contains the data and which fields are needed.

Similar access is available to COBOL, PL/I, C, Assembler, Pascal, and FORTRAN programs using subroutines included with the SHAREFILE/AP product.

FILE SHARING

SHAREFILE allows full file-sharing capabilities, a feaure not generally available in IBM's VM or MVS environments.

No significant limitation exists to the number of users that can access a file at the same time. Easy-to-use control tools allow complete read and write sharing of files. Once you designate certain files as shared files, there is no special action required when others use the files.

SECURITY

Through the use of a central file server, SHAREFILE/AP maintains ownership control over all files, permitting authorized access only.

File security is accomplished using a mechanism called an access table which

is initially empty. Upon creation, a file can be accessed only by its owner. If the owner wishes to allow another user to access that file, an entry must be made in the access table specifying the other user's identification and what types of access will be permitted. Numerous levels of access, such as read, write, and erase can be granted in any combination.

ACCOUNTABILITY

The SHAREFILE/AP System provides file storage and file resource unit accounting information, giving you the opportunity to monitor SHAREFILE/AP System usage at your site, and alternatively, "chargeback" your users for resources consumed.

File storage accounting information reports the total bytes owned by a user. This information is automatically recorded every time the SHAREFILE/AP System is started. File Resource Units (FRUs) are calculated on each file operation performed by a user and represent work done by the SHAREFILE/AP System Server in processing a user's work.

MINIMUM OPERATOR INTERVENTION

The SHAREFILE/AP System is designed to run with a minimum of operator intervention. The system runs as a disconnected virtual machine under VM or as a batch job or started task under MVS. The only operational tasks requiring attention are STARTUP and SHUTDOWN, and running utility programs. A complete set of system utilities is provided with SHAREFILE/AP for such tasks as BACKUP and RESTORE. These backup and restore utilities may be run while the file system is still running.

RELIABILTIY, MAINTAINABILITY, EFFICIENCY

While SHAREFILE is written in PL360 for reliability and maintainability, it is also very efficient. Certain key modules are optimized in 370 Assembly language to ensure superior performance. By design, SHAREFILE is extendable to new I/O devices in a much easier fashion than was possible in earlier APL file systems.

SHAREFILE/AP COMMANDS

Function	Purpose
FAPPEND	Adds a component to the end of a file.
FAVAIL	Determines if the SHAREFILE System is available.
FCREATE	Creates a new file.
FDROP	Drops components from the beginning or end of a file.
FERASE	Erases a file.
FHIST	Provides historical information about a SHAREFILE file.
FHOLD	Holds a file for updating.
FILES	Lists the SHAREFILE files that a user owns.
FLIB	Lists files in a library.
FNAMES	Lists names of tied files.
FNUMS	Lists tie numbers of tied files.
FRDAC	Reads the access table of a file.
FRDCI	Reads information about a component of a file.
FREAD	Reads a component from a file.
FRENAME	Renames a file.
FREPLACE	Replaces a value into an existing component of a file.
FRESIZE	Changes the size limit of a file.
FSIZE	Queries the size of a file.
FSTAC	Sets the access table of a file.
FSTATUS	Determines the status of a particular file (for example, who has the file tied, or who has a file held).
FSTIE	Ties, or activates, a file for shared use.
FSTORAGE	Displays and optionally sets the current storage limit.
FSWITCH	Selects among several SHAREFILE Systems which may be concurrently executing.
FTIE	Exclusively ties, or activates, a file for subsequent use.
FUNTIE	Unties one or more files from the active list.
LHIST	Provides historical library information.
LIBS	Lists the SHAREFILE libraries that a user owns.

COMPATIBILITY

STSC's systems software and applications software are based on over 18 years of development efforts in its core technology--APL. Over the years STSC has improved the performance and extended the range of APL with proprietary systems software packages such as SHAREFILE/AP. Our development has enabled us to offer information processing systems that are general enough to support a wide range of users, yet can be customized more easily than traditional packaged software.

Since STSC introduced the first shared access, component-based APL file system in 1969, it has become the industry standard for state-of-the-art file system design. Over the years, the SHAREFILE System has been adapted to various machine architectures and environments. SHAREFILE is available with STSC's APL*PLUS System running on micro, supermicro, mini, and mainframe computers in the MS-DOS®, UNIX™, VAX/VMS, VM/CMS, and MVS/TSO environments.

COMPREHENSIVE DOCUMENTATION

STSC'S SHAREFILE/AP System is extensively documented in the APL*PLUS SHAREFILE/AP System documentation (STSC, 1988), provided with the software.

AVAILABILITY

The SHAREFILE/AP System may be licensed as a software product for use on your own computer. Contact your STSC Mainframe Software Representative at telephone (301) 984-5000, telefax (301) 984-5094 or at telex account 898085 STSC ROVE for further information.

STSC

STSC, Inc
2115 East Jefferson Street
Rockville, Maryland 20852
(301) 984-5000

APL*PLUS is a registered trademark of STSC, Inc., SHAREFILE is a trademark of STSC, Inc.
APL2 is a registered trademark of International Business Machines
UNIX is a trademark of AT&T Bell Laboratories.
MS-DOS is a registered trademark of Microsoft Corporation
Copyright © 1988, STSC, Inc.

APL★PLUS® UNX System

STSC's APL★PLUS UNX System is the most advanced APL system available for the UNIX* operating environment. Encompassing the best features of the APL★PLUS Systems for mainframes and microcomputers, the APL★PLUS UNX System provides advanced APL technology—catering to today's most powerful micro and minicomputers.

Built in performance-oriented features help you work efficiently. The APL★PLUS UNX System's unique combination of features include partial compilation of APL code, Nested Arrays, and a powerful multiple image ring-editor.

With the APL★PLUS UNX System there are no workspace and individual variable limitations. And, because it can store and access data in disk files outside your APL workspace, the APL★PLUS UNX System is an excellent environment for multi-user applications.

DATA TYPES:

Boolean:	1bit per element
Integer:	32 bits per element
Floating Point:	64-bit IEEE Standard double precision
Character:	8 bits per element
Nested:	32 bit pointer element
Heterogeneous:	mixed datatypes

SYSTEM LIMITS:

Numeric range:	−1.797E308 to 1.797E308
Maximum rank:	127
Maximum variable size:	2,147,483,644 bytes
Maximum coordinate length:	2,147,483,647
Maximum lines in function:	32,767
Maximum symbols in workspace:	32,767
Maximum identifier length:	100 characters
Maximum workspace size:	limited only by UNIX process size limit

SYSTEM FEATURES

- A complete APL★PLUS implementation in an extended working environment.
- Nested arrays compatible with STSC's mainframe APL★PLUS System and IBM's APL2.
- Partial compilation of APL functions on initial execution.
- Mixed APL and ASCII input accepted.
- Workspace size limited only by memory available.
- Built-in terminal emulation and Upload/Download communications capability.
- Full-screen word processing style editor for functions or character data; allows editing of multiple objects at the same time.
- Off-screen scrolling.
- User definable function keys.
- Status line.
- Interfaces to non-APL programs and packages.
- Ability to execute UNIX commands directly.
- Component file system and interfaces to native UNIX files; allows shared read and write access.
- Report-formatting facility.
- Error-trapping features.
- Over 150 system functions.
- Comprehensive documentation.

SYSTEMS FUNCTIONS

APL Component Files:

- ☐ *FAPPEND* —Append array to component file
- ☐ *FAVAIL* —Report file system availability
- ☐ *FCREATE* —Create component file
- ☐ *FDROP* —Drop components from file
- ☐ *FDUP* —Duplicate file and compact unused space
- ☐ *FERASE* —Erase component file
- ☐ *FHIST* —Historical timestamps for file
- ☐ *FHOLD* —Hold file, synchronization by queue
- ☐ *FLIB* —List component files in a directory
- ☐ *FNAMES* —List names of tied files
- ☐ *FNUMS* —List tie numbers of tied files
- ☐ *FRDAC* —Read file access matrix
- ☐ *FRDCI* —Read audit-trail information on file component
- ☐ *FREAD* —Read component from file into workspace
- ☐ *FRENAME* —Rename component file
- ☐ *FRESIZE* —Change size limit on file
- ☐ *FSIZE* —Report file size information
- ☐ *FSTAC* —Set file access matrix
- ☐ *FSTIE* —Share tie (open for concurrent access)
- ☐ *FTIE* —Exclusive tie (non-concurrent access)
- ☐ *FUNTIE* —Untie (close) files

Native UNIX Files:

- ☐ *NAPPEND* — Append APL data to regular UNIX file as byte stream
- ☐ *NCREATE* — Create UNIX file
- ☐ *NERASE* — Erase UNIX file
- ☐ *NNAMES* — List names of tied (open) UNIX files
- ☐ *NNUMS* — List tie numbers of open UNIX files
- ☐ *NRDAC* — Read permission bits on UNIX files
- ☐ *NREAD* — Read data from UNIX file
- ☐ *NRENAME* — Rename UNIX file
- ☐ *NREPLACE* — Replace data in UNIX file
- ☐ *NSIZE* — Report size of UNIX file
- ☐ *NSTAC* — Set permission bits on UNIX file
- ☐ *NTIE* — Open native UNIX file (including TTY port or other UNIX special files)
- ☐ *NUNTIE* — Untie (close) UNIX files

Saved Workspaces:

- ☐ *COPY* — Copy objects under program control
- ☐ *LOAD* — Load workspace under program control
- ☐ *PCOPY* — Protective copy under program control
- ☐ *PSAVE* — Protective workspace save
- ☐ *QLOAD* — Quiet load under program control
- ☐ *SAVE* — Save workspace under program control
- ☐ *WSLIB* — List saved workspaces in a directory
- ☐ *XLOAD* — Load workspace without executing latent expression

Active Workspace Environment:

- ☐ *AI* — Accounting Information
- ☐ *CR* — Canonical representation of a function
- ☐ *CRL* — Canonical representation of one line of a function
- ☐ *CRLPC* — Extract public comment from a function
- ☐ *CT* — Comparison tolerance
- ☐ *DEF* — Define a function from visual form
- ☐ *DEFL* — Replace/insert line in a function
- ☐ *DL* — Delay for specific interval
- ☐ *DR* — Reports internal data type of variable
- ☐ *ERASE* — Erase a function or variable
- ☐ *EX* — Erase a function or variable
- ☐ *FX* — Define a function from canonical form
- ☐ *IDLIST* — List objects in a workspace
- ☐ *IDLOC* — Report definitions of localized identifiers at all stack levels
- ☐ *IO* — Index origin
- ☐ *LC* — Line counter
- ☐ *LOCK* — Lock a function under program control
- ☐ *LX* — Latent expression
- ☐ *NC* — Name class of an APL object
- ☐ *NL* — List objects of a given class in workspace
- ☐ *PP* — Print precision
- ☐ *PR* — Prompt replacement for quote-quad I/O
- ☐ *PW* — Print width
- ☐ *RL* — Random number seed
- ☐ *SI* — Report current state indicator stack
- ☐ *SIZE* — Report size of an APL object
- ☐ *STOP* — Report or set stop vector on a function
- ☐ *SYMB* — Report current symbol table size
- ☐ *TRACE* — Trace function execution
- ☐ *VR* — Visual representation of a function
- ☐ *WA* — Amount of unused workspace area available
- ☐ *WSID* — Name of active workspace
- ☐ *WSSIZE* — Size of current active workspace
- ☐ *WSOWNER* — Identify owner of active workspace
- ☐ *WSTS* — Timestamp of current active workspace

Exception Handling:

- ☐ *ALX* — Latent-expression handler for weak interrupts
- ☐ *DM* — Most recently produced error message
- ☐ *ELX* — Latent-expression handler for APL errors
- ☐ *ERROR* — Signal APL error under program control
- ☐ *SA* — Stop action: clears stack or exits APL on immediate execution input

Formatting:

- □ *FI* — Input conversion of character to numeric
- □ *FMT* — General purpose report formatter
- □ *VI* — Input conversion verifier

Communications and Peripherals:

- □ *ARBIN* — General-purpose input/output function for communicating with peripherals and remote computers. Handles translation, overstriking, time-outs, buffer sizes, and terminator characters
- □ *ARBOUT* — Transmits arbitrary character sequences to user's terminal
- □ *CURSOR* — Move cursor to position on screen or reports current position
- □ *GTTY* — Report information on state of TTY port (baud rate, etc.)
- □ *INKEY* — Read a single keystroke from the terminal
- □ *IOCTL* — Manipulate TTY device driver parameters
- □ *PFKEY* — Read or set logical function (soft) keys
- □ *STTY* — Set TTY port baud rate and other parameters
- □ *TERM* — Terminal emulator for signing on to a remote computer
- □ *WGET* — Get data from terminal screen
- □ *WINDOW* — Report terminal screen dimensions
- □ *WPUT* — Write data to terminal screen

Miscellaneous:

- □ *AV* — Character set constant vector
- □ *EDIT* — Full-screen editor for functions and character data
- □ *SS* — String search function
- □ *SYSID* — APL⋆PLUS System identification
- □ *SYSVER* — APL⋆PLUS System version number
- □ *TCBEL* — Character constant for sounding bell (also □ *TCBS*, □ *TCDEL*, □ *TCESC*, □ *TCFF*, □ *TCLF*, □ *TCNL*, □ *TCNUL*)
- □ *TS* — Current time stamp
- □ *UL* — Report number of users signed on to system
- □ *USERID* — Report user name for current user

Interface to UNIX and non-APL Programs:

- □ *CALL* — Call assembler routine in APL workspace (not in all versions)
- □ *CHDIR* — Change working directory
- □ *LIB* — Report all files in a directory, regardless of type
- □ *LIBD* — Define a library number as a UNIX directory
- □ *LIBS* — Report current definitions of numbered libraries
- □ *SH* — Execute UNIX shell command under program control
- □ *STPTR* — Locate physical address of APL object in workspace
- □ *XP1*, □ *XP2*, □ *XP3*, □ *XP4* and □ *XP5* — Interface to as many as five external programs

System Commands:

-) — Recall last APL statement for editing
-)*CLEAR* — Clear workspace, optionally changing workspace size
-)*COPY* — Copy from saved workspace
-)*DROP* — Delete saved workspace
-)*EDIT* — Use full-screen editor to edit function or variable
-)*ERASE* — Erase APL objects
-)*FLIB* — List component files in a directory
-)*FNS* — List functions in workspace
-)*LIB* — List all files in a directory
-)*LOAD* — Load a saved workspace
-)*OFF* — End APL session
-)*PCOPY* — Copy from saved workspace without replacement
-)*PORTS* — List users signed on to UNIX
-)*PSAVE* — Save workspace without replacement
-)*RESET* — Clear state indicator

)*SAVE*	— Save the active workspace
)*SH*	— Execute UNIX shell command
)*SI*	— Display state indicator
)*SIC*	— Clear state indicator
)*SINL*	— Display state indicator and localized names
)*SYMBOLS*	— Display or change size of symbol table
)*VARS*	— List variables in a workspace
)*WSID*	— Display or change name of active workspace
)*WSLIB*	— List workspace in a directory
)*XLOAD*	— Load workspace suppressing latent expression

SYSTEM REQUIREMENTS

The APL★PLUS UNX System is designed for computers that support the UNIX Operating Systems. A minimum of 512K bytes of main memory is required, and 1024K bytes is recommended.

PRODUCT CONTENTS

The APL★PLUS UNX System consists of:

- Complete documentation in custom linen binders and boxes
- System software on tape or disk
- An APL tutorial

PORTABILITY

The APL★PLUS UNX System is compatible with STSC's other APL★PLUS Systems running in the TSO, CMS, and the PC DOS environments. Built-in communications software allows the transfer of data and programs to and from computers.

AVAILABILITY

STSC's APL★PLUS Systems are available through local dealers or by contacting STSC, Inc. An APL★PLUS UNX Run-Time System is available for application developers.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

APL★PLUS is a registered trademark of STSC, Inc. PLUS★WARE is a trademark of STSC, Inc. UNIX is a trademark of AT&T Bell Laboratories.

Copyright © 1986, STSC, Inc.
Printed in the U.S.A.
CC190-0286

A PLUS★WARE™ PRODUCT

THE APL★PLUS® UNX SYSTEM

Enter the ultimate APL dimension with STSC's APL★PLUS UNX System—the most advanced APL system available today. STSC has refined APL★PLUS technology to work under the UNIX™ operating systems. The APL★PLUS UNX System encompasses the best features of APL★PLUS Systems for mainframes and microcomputers, and adds the latest in software technology.

PLUG IN HIGH PERFORMANCE APL

Once you try it, you'll see that the APL★PLUS System out-performs any other APL system designed for microcomputers. That's because we've built in performance-oriented features to help you work efficiently.

For instance, the system partially compiles any new line of APL code into an internal "pseudo-code." Unlike most APL systems, this pseudo-code needs no repeated syntactic analysis—so it doesn't slow you down.

This feature gives interpreted APL the speed benefits of compiled languages, so you can enjoy the flexibility and productivity of APL along with the machine efficiency of compiled code.

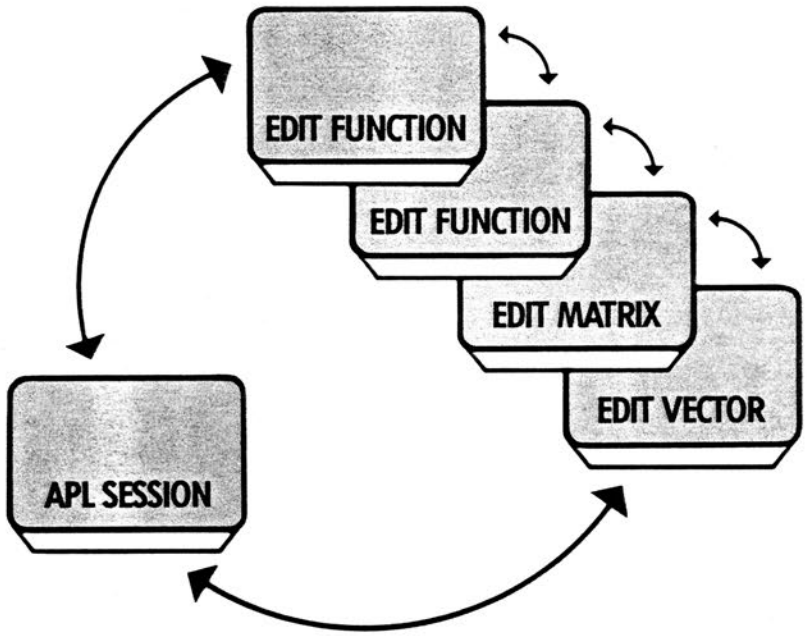
BEYOND FULL-SCREEN EDITING

The APL★PLUS System takes you beyond traditional full-screen editing. Create, view and edit as many images as you want—simultaneously! Create images within your editor—or copy them from functions and variables in your workspace.

Although only one image shows on your screen at a time, you can work on any number in a single session. For ease of tracking, the number and label of the image you are currently working on appears on the status line. Images are numbered in a ring, so you can move from one to another in either direction. The session itself is also an object in the ring, allowing you to freely copy text from the editor to the active workspace, or vice versa.

DISCOVER THE UNLIMITED WORKSPACE

The APL★PLUS UNX System frees you from the limitations of other APL systems for small computers. Workspaces and individual variables can be as large as your computer allows—up to 2 billion bytes! Arrays can have as many as 214 million elements and 127 dimensions. Integer data is stored in 32 bits, just like on a mainframe, with boolean data packed 1 element per bit.



SHARE DATA WITH COLLEAGUES
You'll find the APL★PLUS UNX System an excellent environment for multi-user applications because it can store and access data in disk files outside your APL workspace. The system lets you share APL★PLUS component files with your colleagues, plus handle other files native to the UNIX operating system. True concurrent file sharing lets users safely update component files at the same time.

INTERFACE WITH OUTSIDE PROGRAMS
Our external process interface lets you use the APL★PLUS UNX System with a wide variety of non-APL programs: database managers, full-screen forms-input processors, graphics packages and others. And, when necessary, you can quickly call up special-purpose subroutines in compiled languages like C or Pascal.

ADD DIMENSION TO YOUR DATA
In typical applications, we know real data does not always fit neatly into the rectangular arrays of ordinary APL. So we created nested arrays—arrays combining items of different dimensions to show data in its varied natural shapes. Using the varied dimensions of nested arrays will help you write simpler, faster, more bug-free programs.

The APL★PLUS UNX System nested arrays are identical to those in our APL★PLUS Systems for mainframes, and compatible with IBM's APL2.

HOOK UP TO MORE CAPABILITY
Use the APL★PLUS UNX System's flexible communications facility to log onto a remote host computer and transfer data, and take advantage of the same features to drive pen plotters or other peripheral printers. We provide tools to make it easy to transfer data and programs between the APL★PLUS UNX System and other APL★PLUS Systems, such as our APL★PLUS PC System and APL★PLUS Mainframe System.

BULLET-PROOF YOUR APPLICATIONS
Error-handling features mean that applications can act friendly, even when users make mistakes. Our exception handling features allow applications to react automatically to errors and exceptions that occur during program execution, to take corrective action, or simply keep the program from terminating. Exception handling can also improve the security of applications, protecting programs and data from accidental or malicious interference.

CREATE REPORTS WITH EASE
A general-purpose formatting capability provides a simple yet powerful method for producing attractive reports. STSC's well-known report formatting facility combines the best of FORTRAN and COBOL picture editing to make any data formatting task easy.

AVAILABILITY
The APL★PLUS UNX System is available on a wide variety of UNIX computers, from the IBM PC/AT® to the DEC® VAX® family.

The product may be purchased through local dealers or by contacting STSC, Inc. Call us today for more information or to discuss your personal APL needs.

APL★PLUS® System for VAX/VMS

STSC'S APL★PLUS System is an advanced APL system for the DEC VAX under the VMS operating system. Encompassing the best features of the APL★PLUS System for mainframes and microcomputers, the APL★PLUS System for VAX/VMS provides advanced APL technology for Digital's full family of powerful micro, mini, and mainframe VAX computers.

Built-in performance-oriented features help you work efficiently. The APL★PLUS System's unique combination of features include partial compilation of APL code, nested arrays, and a powerful full-screen editor that combines the session manager with an editor for APL functions and variables.

With the APL★PLUS System there are no workspace or variable limitations. And, because it can store and access data in VMS files outside your APL workspace, the APL★PLUS System is an excellent environment for multi-user applications.

DATA TYPES:

Boolean:	1 bit per element
Integer:	32 bits per element
Floating Point:	64 bit VAX double precision
Character:	8 bits per element
Nested:	32 bit pointer element
Heterogeneous:	Mixed datatypes

SYSTEM LIMITS:

Numeric range:	−1.7014E38 to 1.7014E38
Maximum rank:	127
Maximum size:	2,147,483,644 bytes
Maximum coordinate length:	2,147,483,647
Maximum lines:	32,767
Maximum symbols in workspace:	32,767
Maximum identifier length:	100 characters
Maximum workspace size:	available memory

SYSTEM FEATURES:

- A complete APL★PLUS System implementation in an extended working environment.
- Nested arrays compatible with STSC's mainframe APL★PLUS System and IBM's APL2.
- Partial compilation of APL functions on initial execution, avoiding repeated syntax analysis.
- Mixed APL and ASCII input accepted.
- Workspace size limited only by memory available.
- Built-in terminal emulation and upload/download communications capability.
- Full-screen editor for functions or character data allows editing of multiple objects at the same time.
- Session manager with off-screen scrolling.
- User-definable function keys.
- Status line.
- Interfaces to non-APL programs and packages.
- Ability to execute VMS DCL commands directly.
- Component file system and interface to VMS files allowing shared read and write access.
- Report-formatting facility.
- Error-trapping features.
- Over 150 system functions.
- Comprehensive documentation.

SYSTEM FUNCTIONS

APL Component Files:

☐ *FAPPEND*	Append array to component file
☐ *FAVAIL*	Report file system availability
☐ *FCREATE*	Create components from file
☐ *FDROP*	Drop components from file
☐ *FDUP*	Duplicate file and compact unused space
☐ *FERASE*	Erase component file
☐ *FHIST*	Historical timestamps for file
☐ *FHOLD*	Hold file, synchronization by queue
☐ *FLIB*	List component files in a directory
☐ *FNAMES*	List names of tied files
☐ *FNUMS*	List tie numbers of tied files
☐ *FRDAC*	Read file access matrix
☐ *FRDCI*	Read audit trail information on component
☐ *FREAD*	Read component from file into workspace
☐ *FRENAME*	Rename component file
☐ *FREPLACE*	Replace a component with a new value
☐ *FRESIZE*	Change size limit on file
☐ *FSIZE*	Report file size information
☐ *FSTAC*	Set file access matrix
☐ *FSTIE*	Share tie (open for concurrent access)
☐ *FTIE*	Exclusive tie (nonconcurrent access)
☐ *FUNTIE*	Untie (close) files

Native VMS Files:

□ *LIB*	List names of files in a directory
□ *NAPPEND*	Append APL data to native VMS file
□ *NCREATE*	Create VMS file
□ *NERASE*	Erase VMS file
□ *NNAMES*	List names of tied (open) VMS files
□ *NNUMS*	List tie numbers of open VMS files
□ *NRDAC*	Read permission bits on VMS files
□ *NREAD*	Read data from file
□ *NRENAME*	Rename file
□ *NREPLACE*	Replace data in file
□ *NSIZE*	Report size of file
□ *NSTAC*	Set permission bits on file
□ *NTIE*	Open native file
□ *NUNTIE*	Untie (close) files

Saved Workspaces:

□ *COPY*	Copy objects under program control
□ *LOAD*	Load workspace under program control
□ *PCOPY*	Protective copy under program control
□ *PSAVE*	Protective workspace save
□ *QLOAD*	Quiet load under program control
□ *SAVE*	Save workspace under program control
□ *WSID*	List saved workspaces in a directory
□ *XLOAD*	Load workspace without executing □*LX*

Active Workspace Environment:

□ *AI*	Accounting information
□ *CR*	Canonical representation of a function
□ *CRL*	Canonical representation of function line
□ *CRLPC*	Extract public comment from a function
□ *CT*	Comparison tolerance
□ *DEF*	Define a function from visual form
□ *DEFL*	Replace/insert a line in a function
□ *DL*	Delay for specific interval
□ *DR*	Report internal data type of variable
□ *ERASE*	Erase a function or variable
□ *EX*	Erase a function or variable
□ *FX*	Define a function from canonical form
□ *IDLIST*	List objects in a workspace
□ *IDLOC*	Report definitions of localized identifiers
□ *IO*	Index origin
□ *LC*	Line counter
□ *LOCK*	Lock a function under program control
□ *LX*	Latent expression
□ *NC*	Name class of an APL object
□ *NL*	List of workspace object names
□ *PP*	Print precision
□ *PR*	Prompt replacement for ⍞ I/O
□ *PW*	Print width
□ *RL*	Random number seed
□ *SI*	Report current state indicator stack
□ *SIZE*	Report size of an APL object
□ *STOP*	Report or set a stop vector for a function
□ *SYMB*	Report current symbol table size
□ *TRACE*	Trace function execution
□ *VR*	Visual representation of a function
□ *WA*	Amount of unused workspace available
□ *WSID*	Name of the active workspace
□ *WSSIZE*	Size of current workspace
□ *WSOWNER*	Identify owner of active workspace
□ *WSTS*	Timestamp of current active workspace

Exception Handling:

□ *ALX*	Latent expression handler for weak interrupts
□ *DM*	Most recently produced error message
□ *ELX*	Latent expression handler for APL errors
□ *ERROR*	Signals APL error under program control
□ *SA*	)*CLEAR* or)*OFF* on immediate execution

Formatting:

□ *FI*	Input conversion of character data
□ *FMT*	General purpose report formatter
□ *VI*	Input conversion verifier

Communications and Peripherals:

□ *ARBIN*	General purpose input/output function for communicating with peripherals and remote computers. Handles translation, overstriking, timeouts, buffer size, and terminator characters.
□ *ARBOUT*	Send arbitrary character sequences to terminal
□ *CURSOR*	Move cursor or report cursor position
□ *INKEY*	Get a single keystroke from the terminal
□ *PFKEY*	Set or query a pseudo function key definition
□ *WGET*	Get data from terminal screen
□ *WINDOW*	Report terminal screen dimensions
□ *WPUT*	Write data to terminal screen

Miscellaneous:

□ *AV*	Character set constant vector
□ *EDIT*	Full-screen editor for functions and character data
□ *MF*	Execution time monitor facility
□ *SS*	String search function
□ *SYSID*	APL★PLUS System identification
□ *SYSVER*	APL★PLUS System version number
□ *TC*$_{xxx}$	Character constants (□ *TCBEL*, □ *TCBS*, □ *TCDEL*, □ *TCESC*, □ *TCFF*, □ *TCLF*, □ *TCNL*, □ *TCNUL*)
□ *TS*	Current timestamp
□ *UL*	Report number of users signed on to the system
□ *USERID*	Report user name for the current user

Interface to Non-APL Programs:

□ *CMD*	Execute DCL commands for APL
□ *LIBD*	Access directories as library numbers
□ *CHDIR*	Change the working directory
□ *NA*	Run non-APL programs from APL
□ *XPn*	Communicate with an external process

System Commands:

)	Recall last APL statement
)*CMD*	Execute a DCL command
)*CLEAR*	Clear the workspace
)*COPY*	Copy from saved workspace
)*DROP*	Delete saved workspace
)*EDIT*	Edit function or variable with full-screen editor
)*ERASE*	Erase APL objects
)*FLIB*	List component files in a directory
)*FNS*	List all functions in a workspace
)*LIB*	List all files in a directory
)*LOAD*	Load a saved workspace
)*OFF*	End an APL session
)*PCOPY*	Copy from a saved workspace without replacement
)*PORTS*	List users signed on to VMS
)*PSAVE*	Save workspace without replacement
)*RESET*	Clear state indicator
)*SAVE*	Save the active workspace
)*SI*	Display the state indicator stack contents
)*SIC*	Clear the state indicator stack
)*SINL*	Display the state indicator with localized names
)*SYMBOLS*	Display or change the size of the symbol table
)*VARS*	List variables in the workspace
)*WSID*	Display or change the name of the active workspace
)*WSLIB*	List workspaces in a directory
)*XLOAD*	Load a workspace without executing □ *LX*

SYSTEM REQUIREMENTS

The APL★PLUS System for VAX/VMS is designed for all of Digital's VAX computers with the VMS operating system Version 4.0 or later.

PRODUCT CONTENTS

The APL★PLUS System for VAX/VMS consists of:

- System software on tape or cartridge
- Installation Guide
- The APL tutorial *APL is Easy!*
- Informative User's Manual
- Comprehensive Reference Manual

PORTABILITY

The APL★PLUS System for VAX/VMS is compatible with STSC's APL★PLUS System for the MVS/TSO, VM/CMS, UNIX, PC-DOS and Macintosh environments. Built-in communications software makes it easy to transfer data and programs to and from various computer systems.

AVAILABILITY

STSC's APL★PLUS System is available through STSC or local dealers outside the U.S. A run-time version of the software is available for application developers with APL-based software products.

STSC—THE APL COMPANY!

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 638-6660
(301) 984-5000 (In Maryland or Canada)

APL★PLUS is a registered trademark and service mark of STSC, Inc.
PLUS★WARE is a registered trademark of STSC, Inc.
VAX and VMS are trademarks of Digital Equipment Corporation. UNIX is a trademark of AT&T Bell Laboratories.
Apple Macintosh is a trademark of Macintosh Laboratories, Inc.

Copyright © 1987, STSC, Inc.
Printed in the U.S.A.

A PLUS★WARE® PRODUCT

APL★PLUS® System for VAX/VMS

STSC

Introducing STSC's High Performance APL System for VAX/VMS

Once you try it, you'll see that the APL⋆PLUS System is built for speed and productivity. That's because we've built in performance-oriented features to help you and your computer work efficiently.

The APL⋆PLUS System partially compiles any new line of APL code into an internal "pseudo-code." Unlike most APL systems, this pseudo-code needs no repeated syntactic analysis—so it doesn't slow you down.

The APL⋆PLUS System can use all the available memory in your VAX to tackle significant problems that require huge workspaces and arrays. The system is designed so you won't be fenced in by arbitrary limits.

The new monitor facility can lead you straight to the hot spots in your code where optimization is needed. This means you can develop faster programs in less time.

Interface with Outside Programs

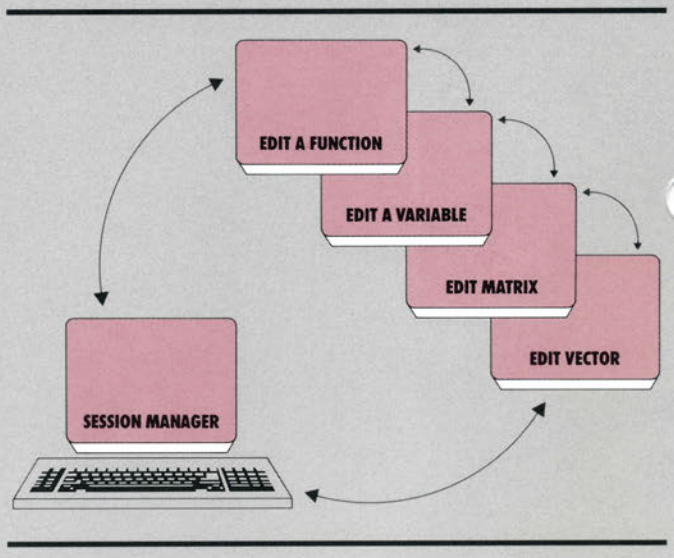
Our external process interfaces allow you to use the APL⋆PLUS System with a wide variety of non-APL programs: database managers, full-screen forms-input processors, graphics packages and others.

With ⎕NA, you can quickly call up subroutines in compiled languages like C or FORTRAN as if they were regular APL functions. And, with ⎕CMD and)CMD, you can also execute VMS DCL commands from the APL environment.

Enjoy a Full-Screen Editor for the VAX

The APL⋆PLUS System creates a full-screen environment for the most popular VAX terminals including VT220 and 240, as well as a host of common APL terminals, PC's, and PC-compatibles running APL⋆PLUS.

Our full-screen session manager boosts your productivity by keeping a record of the APL session so you can recall and re-execute earlier statements. You can create, view and edit multiple functions and variables—simultaneously! The session manager is integrated with the full-screen editor, providing a seamless environment for developing, testing and using APL programs.



Although only one image shows on your screen at a time, you can edit any number in a single session. For ease of tracking, the number and label of the image you are currently working on appear on the status line. Images are numbered in a ring, so you can move quickly from one to another in either direction. The session itself is also an object in the ring, allowing you to freely copy text from the editor to the active workspace, or vice versa.

Add Dimension to Your Data
In typical applications, real data does not always fit neatly into the rectangular arrays required by ordinary APL. So we designed the APL★PLUS System for VAX/VMS to include nested arrays—arrays in which each item of the array can itself be an array of any size, shape, or data type. This allows you to store and display data in a simpler, more natural form. Moreover, we added powerful generalized operators such as "each" to let you replace function loops with expressions that take only a few keystrokes—so you can write simpler, faster, more bug-free code.

STSC released the world's first nested arrays system in 1981. Our extensive experience with second generation APL has allowed us to design the APL★PLUS System for VAX/VMS to deliver exceptional performance in nested arrays.

Share Data
You'll find the APL★PLUS System an excellent environment for multi-user applications because it lets you share component files with others. True concurrent file sharing allows users to safely update component files at the same time.

In addition, you can handle files native to the VMS operating system. Storing and accessing data in disk files outside your APL environment allows you to exchange data easily with other software packages.

Create Reports with Ease
A general-purpose formatting capability provides a simple yet powerful method for producing attractive reports. STSC's well-known report formatting facility combines features of FORTRAN and COBOL picture editing to make any data formatting task easy.

Bullet-proof Your Applications
Error-handling features mean that applications can act friendly, even when users make mistakes. Our exception handling features allow applications to react automatically to errors and exceptions that occur during program execution, to take corrective action, or simply keep the program from terminating. Exception handling can also improve the security of applications, protecting programs and data from accidental or malicious interference.

COMPATIBILITY

The APL★PLUS System is available for the full line of VAX/VMS computers, from the MicroVAX II through the 8800 series of superminis and mainframes. But the APL★PLUS System is also compatible across a wide range of other hardware environments and operating systems including:

- IBM mainframes (VM and MVS)
- IBM 9370
- IBM PC and PC-compatibles
- Apple Macintosh
- Unix-based machines

In addition, our users can take advantage of the APL★PLUS System's communications facilities to transfer data and programs to and from other APL★PLUS Systems on minis, micros, and mainframes.

As hardware costs decrease and manpower costs increase, you need the flexibility to use the most cost-effective hardware that meets your needs—without rewriting your software. APL★PLUS is the only APL system available today that does not limit you to a single family of computers. As your computing requirements change, STSC can provide an APL★PLUS System that meets your needs.

AVAILABILITY

STSC's APL★PLUS System is available through STSC or local dealers outside the U.S. A run-time version of the software is available for application developers with APL-based software products.

COMMITMENT

APL Is Our Business
STSC's commitment to APL has set the standard for the APL of today and tomorrow. We are committed to provide you with the most comprehensive, performance-oriented APL in the industry—now and in the future.

Experience
More than 18 years of experience in the development and delivery of premium-quality APL products and services... More than 20,000 APL★PLUS installations, to assure you of proven, reliable software.

Technology
Ongoing enhancements which incorporate the best new technologies to improve the quality, performance and value of your APL★PLUS System.

Support
- Telephone support
- Comprehensive maintenance plan including system upgrades
- Complete product documentation

Service
Full program of support services to meet your special requirements.

- Professional APL consulting staff
- Training classes
- APL publications

STSC—THE APL COMPANY!

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 638-6660
(301) 984-5000 (In Maryland or Canada)

APL★PLUS is a registered trademark and service mark of STSC, Inc.
PLUS★WARE is a resitered trademark of STSC, Inc.
VAX and VMS are trademarks of Digital Equipment Corporation. UNIX is a trademark of AT&T Bell Laboratories.
IBM is a trademark of International Business Machines.
Apple Macintosh is a trademark of Macintosh Laboratories, Inc.

Copyright © 1987, STSC, Inc.
Printed in the U.S.A.

A PLUS★WARE® PRODUCT

STSC's
PLUS★WARE™ Products

APL★PLUS®/PC
FINANCIAL AND STATISTICAL LIBRARY

APL★PLUS/PC Financial and Statistical Library

Successful financial and statistical modeling and analysis invariably relies on rapidly collecting and processing diverse information. The results must be timely, accurate, and complete. And the tools you use will determine your success. That's why you should use the APL★PLUS/PC Financial and Statistical Library.

The Library is an extensive collection of more than 200 APL programs that forms a powerful and comprehensive set of capabilities, specially prepared to work with STSC's highly acclaimed APL★PLUS/PC System. You can use the programs to:

- perform financial calculations
- manipulate and forecast time series
- compute statistical measures
- generate probability distribution samples.

The Financial and Statistical Library was developed by and for professional analysts like yourself. The Library contains the same proven routines that have been used for over a decade in STSC's premier Financial Planning System (FPS).

Illustrative of the depth of the Library is its incorporation of five different types of loan amortization, six types of depreciation, eight types of rate of return, more than fifteen types of forecasting, and more than thirty probability distributions. And most programs routinely accept parameters (for example, data periodicity) and switches (for example, payments in advance or in arrears) that further specify the computations to be performed.

The Library can be used directly as a stand-alone financial or statistical calculator for rapid, ad hoc investigations. Or it can be used as a cohesive foundation for building financial, statistical, or decision-support applications.

The Financial and Statistical Library affords you quick access to its routines in a style exactly matched to the formality of the analysis you want to perform. It supplies you with reliable software encompassing thousands of parametric variations. And it allows you to concentrate your system building efforts on application specific user interfaces rather than the programming of sophisticated algorithms.

The combination of the Financial and Statistical Library and the APL★PLUS/PC System is the complete solution to your analytic needs—a single environment that includes:

- a full-screen editor
- graphics
- communications
- file management
- report formatting
- 8087 co-processor support
- a comprehensive library of financial, forecasting, and statistical routines.

Financial Capabilities

- Bond price, yield, and amortization
- Asset management (cash position, plant requirements)
- Cash flow computations
- Discounted cash flow, net present value, future value
- Discrete or continuous flows, or combinations of both
- Internal rates of return and internal discount factors
- Loan amortization schedules (level payment, variable payment, equal principal, variable principal, term loan)
- Comparison of two level payment mortgage loans
- Depreciation (ACRS, declining balance, sum-of-years' digits, straight line, sinking fund, accelerated sinking fund)
- Leads, lags, and aging schedules
- Interest rates and secondary interest effects
- Inventory costs (LIFO, FIFO, average cost)
- Probabilistic discounted cash flow, present value, future value
- Federal income tax and investment tax credit
- Annuities (present value, future value, periodic balances)

Forecasting Capabilities

- Forecasting by projecting historical values (linear trend, quadratic trend, polynomial trend, step trend, compound growth)
- Forecasting based on historical relationships (regression, average difference, trend of ratio, average ratio)
- Forecasting by parametric methods (S-curve, asymptotic curve, exponential growth, compound growth, polynomial growth)
- Single, double, and triple exponential smoothing
- Weighted and moving averages
- Seasonal adjustment of a time series (regression technique)

Probabilistic Sampling Capabilities

- Empirical probability distributions (discrete, histogram, polygon)
- Analytic probability distributions (Beta, Cauchy, Exponential, Gamma, Geometric, Hypergeometric, Laplace, Linear, Logistic, Lognormal, Normal, Pascal, Triangular, Weibull, and so on)

Statistical Capabilities

- Descriptive statistics (mean, standard deviation, median, mode(s), and so on)
- Frequency distributions for discrete and continuous samples
- Autocorrelation and autocovariance
- Simple, partial, and cross correlation
- Single, multiple, and stepwise regression
- Analysis of regression residuals

And many other capabilities,

including:

- Arithmetic computations with time series
- Structural manipulation of data arrays
- Periodicity conversion for time series (4 reduction methods and 6 expansion methods)
- Range testing of data

Product Contents

The Financial and Statistical Library consists of one single-sided diskette containing the program software, and a comprehensive Reference Manual that describes all of the programs and provides examples of the various computations. The Manual comes in an attractive linen binder with slipcase.

System Requirements

The APL★PLUS/PC Financial and Statistical Library is designed to run under any release of the APL★PLUS/PC System. To use the Library, a minimum of 256 kilobytes of random access memory (RAM) and one disk drive is recommended.

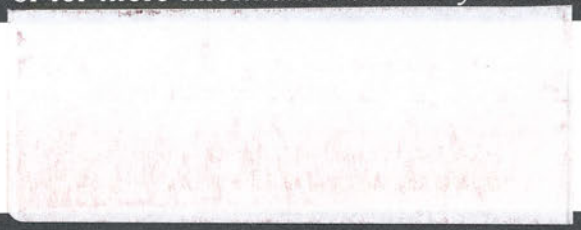
STSC: The Company

STSC, Inc., a Contel Company, designs and implements computer systems for the Information Center environment, manufacturing and distribution planning, and funds management. As an industry leader in the development of the APL programming language, STSC delivers APL-based software and services in the microcomputer and mainframe environments, and publishes APL-based software written by independent developers. STSC has branch offices in major U.S. cities, operates a subsidiary company in the United Kingdom, and licenses software through distributors in several foreign countries.

APL★PLUS is a service mark and trademark of STSC, Inc., registered in the United States Patent and Trademark Office and in certain other countries. PLUS★WARE is a trademark of STSC, Inc.

Copyright © 1984 STSC, Inc.
Printed in the U.S.A.
CC186-0584

To order or for more information contact your local dealer



or call STSC at (800) 592-0050

STSC
A Contel Company

STSC's
PLUS★WARE™ Products

APL★PLUS®/PC TOOLS
Volume 1

APL★PLUS TOOLS Product Line

Productivity is the critical success factor for application developers. Using the APL★PLUS System is a recognized first step towards improving software development productivity. The next most natural step for productivity gain is the acquisition of utility software. STSC satisfies this ongoing need by offering APL★PLUS TOOLS, an extensive library of standardized, consistent, and well-documented development aids and utility subsystems. These programming tools consist of efficient implementations of commonly-required algorithms, integrated application building blocks, interfaces to popular microcomputer hardware and software, utilities to maintain software configurations, and even a selection of amusements that afford the developer an occasional break. You will find that the APL★PLUS TOOLS will:

- speed the progress of your system building efforts,
- expand the capabilities of your microcomputer applications, and
- standardize approaches to common application development requirements.

APL★PLUS TOOLS are released in periodic volumes for APL★PLUS Systems developed for small computers. The tools are derived from STSC's extensive library of proven mainframe and microcomputer software, the same productivity tools that STSC itself has employed for more than a decade, and from submissions to STSC's Software Publishing Program.

APL★PLUS/PC TOOLS, Volume 1

The first volume of tools for the APL★PLUS/PC System offers capabilities in the following areas:

- Communications
- Screen Management
- Reporting and Output Formatting
- Disk and File Management
- Software Development
- Games and Recreation.

COMMUNICATIONS

3278 Terminal Emulator – A customized version of the terminal emulator which runs on the IBM® PC using the IRMA™ Board interface. This software permits the use of a PC as a 3278 Model 2 terminal for an IBM mainframe computer. The keyboard usage of the emulator closely resembles the keyboard usage in the APL★PLUS/PC System.

3278 File Transfer – Software to transfer data rapidly between a PC and a VS APL or APL★PLUS mainframe environment by means of the IRMA Board communications link. Individual variables or entire files can be transferred.

APL-to-IRMA Board Interface – APL functions that interact with the IRMA Board and permit the control of full-screen mainframe applications from the APL★PLUS System on the PC.

PC as Host Computer – Software that allows you to set up one PC as a host computer whereby APL applications can be run from a remote terminal, even through a dial-up telephone connection.

PC-to-PC File Transfer – Software to transfer data from PC to PC after one of them has been established as a host computer. Individual variables or entire files can be transferred.

SCREEN MANAGEMENT

Menu Processing – Programs to present selection menus to, handle interactions with, and interpret responses of an application user.

Menu Maintenance – Programs that allow you to build, edit, maintain, and prepare for instant display menu definitions encompassing an extensive range of features.

Screen Input Facility – Programs to present input forms to, handle interactions with, and interpret and verify responses of an application user.

Screen Forms Maintenance – Programs that allow you to build, edit, maintain, file, and prepare for rapid display screen input form definitions.

REPORTING AND OUTPUT FORMATTING

Output Formatting – A suite of APL subroutines that allows you to direct output to screen/printer, APL component file, or native DOS file; to specify page numbering, titles, footnotes, and top and bottom margins; to group related output on a page; to perform necessary character translations; and to restart interrupted reports.

DISK AND FILE MANAGEMENT

Disk Emulator – A software device driver used to reserve a portion of the computer's random access memory (RAM) so that it appears to DOS 2.00 to be an additional disk storage device, thus permitting extremely fast disk operations. In addition, APL functions to customize the size and RAM address to the device driver are included.

SOFTWARE DEVELOPMENT

Function Maintenance – Utilities to analyze, modify, and maintain individual APL functions.

Workspace Maintenance – Utilities to search, analyze, modify, and maintain a workspace environment.

File Image of a Workspace – A utility to capture a file image of a workspace for configuration or backup schemes or for temporary analysis.

Workspace Re-creation – A utility to re-create a workspace from a file image.

Workspace-file Maintenance – Utilities to search, analyze, modify, and maintain a file image of a workspace.

Workspace Documentation – Software to document fully the contents of a workspace from its file image. A variety of identifier cross-reference and function dependency reports are optional.

Comparison of Workspaces – Software to examine the functions and variables in file images of two workspaces and to display line-by-line differences. Individual pairs of functions, in workspace or file, can also be compared.

GAMES AND RECREATION

Reversi – The classic board game of strategy and capture in a brilliant computer rendition, for one or two players.

Three-dimensional Tic-Tac-Toe – Another favorite board game for solitaire play against the computer.

PRODUCT CONTENTS

Volume 1 of the APL★PLUS/PC TOOLS consists of three diskettes (six single sides) containing the program software, and a comprehensive Reference Manual in an attractive linen binder with slipcase.

SYSTEM REQUIREMENTS

The APL★PLUS/PC TOOLS volumes are designed to operate under the APL★PLUS/PC System, Release 3.0 or later, and under IBM's DOS, version 2.00 or later. Color monitors are supported but not required. A minimum of 256 kilobytes of RAM is suggested although many tools will operate with less.

Use of the IRMA support software in the APL★PLUS/PC TOOLS requires an installed IRMA Board and accompanying software, not included in the APL★PLUS/PC TOOLS.

STSC: The Company

STSC, Inc., a Contel Company, designs and implements computer systems for the Information Center environment, manufacturing and distribution planning, and funds management. As an industry leader in the development of the APL programming language, STSC delivers APL-based software and services in the microcomputer and mainframe environments, and publishes APL-based software written by independent developers. STSC has branch offices in major U.S. cities, operates a subsidiary company in the United Kingdom, and licenses software through distributors in several foreign countries.

APL★PLUS is a service mark and trademark of STSC, Inc., registered in the United States Patent and Trademark Office and in certain other countries. PLUS★WARE is a trademark of STSC, Inc. IBM is a registered trademark of International Business Machines Corporation. IRMA is a trademark of Digital Communications Associates, Inc.

Copyright © 1984 STSC, Inc.
Printed in the U.S.A.
CC187-0584

To order or for more information contact your local dealer

or call STSC at (800) 592-0050

STSC
A Contel Company

The APL★PLUS® SPREADSHEET MANAGER for the PC

Now, powerful spreadsheet capabilities for the APL★PLUS PC environment!

STSC's APL★PLUS Spreadsheet Manager provides you with a powerful new way to prepare and analyze APL data. In addition to enjoying standard spreadsheet features in the APL environment, you can also quickly and easily move date between APL workspaces and LOTUS® or ASCII files.

This spreadsheet capability complements the APL environment so you can solve a wider range of problems in APL faster and easier than ever before. With the Spreadsheet Editor, you can create and analyze tabular data interactively using powerful APL expressions, and make difficult report formatting a thing of the past as you intermix numbers and text with just a few commands.

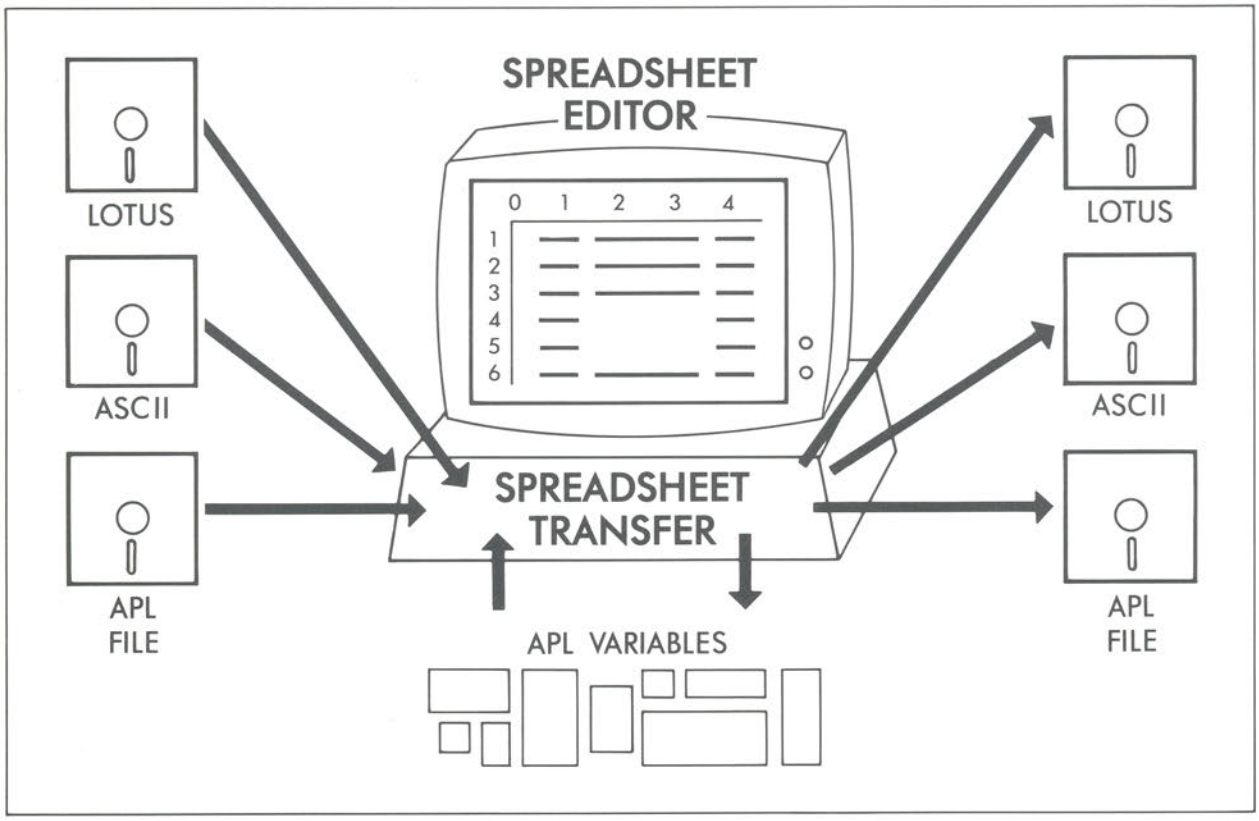
All these capabilities are enhanced by the ability to interface your APL applications to LOTUS (1-2-3® and Symphony™) worksheets and the ASCII files produced by popular database managers, word processors, and other spreadsheet packages. Moving data in and out of these packages is simple with the Spreadsheet Manager—you won't have to re-enter data or write complicated programs to get data in the right format. The Spreadsheet Manager does all of this for you.

THE SPREADSHEET EDITOR

This fast, easy-to-use full-screen spreadsheet editor provides maximum flexibility in analyzing data. The spreadsheet editor is an extension of the full-screen and numeric editors provided with the APL★PLUS PC System.

With the Spreadsheet Manager you can:

- Create APL spreadsheets composed of both character and numeric data; and enter, modify, and review them with a flexible spreadsheet-like editor.
- Execute and store APL formulas for individual cells or ranges of cells in a spreadsheet.
- Move, copy, add, and delete areas of the spreadsheet, and store portions of the spreadsheet in workspace variables. The effect of the last deletion can be reversed by using the editor's cancel (undo) capability.
- Left-justify, center, or right-justify labels.
- Sort columns or rows of the spreadsheet.
- Store and retrieve an APL spreadsheet in a specially



structured APL*PLUS PC System component file.
- Use the APL spreadsheet editor to create, review, or modify arbitrary LOTUS worksheets and ASCII files.

THE TRANSFER FACILITY

The transfer facility provides special capabilities to handle LOTUS 1-2-3 and Symphony worksheets or ASCII text files.

With the Transfer Facility you can:

- Transfer data from LOTUS spreadsheets or ASCII files into APL variables or an APL spreadsheet file.
- Transfer data contained in an APL spreadsheet or in APL variables to external sources, such as LOTUS worksheets or ASCII files.
- Embed non-interactive transfer facilities within other APL*PLUS PC applications.

SYSTEM REQUIREMENTS

The APL*PLUS Spreadsheet Manager is designed for the APL*PLUS PC System, Release 5 or later and an IBM®-PC compatible computer. (Note that although the APL*PLUS PC System is available for some non-IBM compatible computers, the Spreadsheet Manager may not work on machines that are not screen and keyboard compatible with the IBM PC.) A minimum of 384K of RAM is required, with additional memory recommended to facilitate transferring large files and editing large spreadsheets. The APL*PLUS Spreadsheet Manager will handle LOTUS 1-2-3 (Release 1, 1A, and 2), and Symphony (Release 1.0 and 1.1) worksheets.

CONTENTS

The APL*PLUS Spreadsheet Manager software package consists of:

- System software on one two-sided diskette.
- Complete reference documentation in a custom binder.
- Online documentation designed to guide you through the software.
- Sample spreadsheet files and a tutorial.

AVAILABILITY

STSC's APL*PLUS Spreadsheet Manager is available through local dealers or by contacting STSC, Inc.

The APL*PLUS Spreadsheet Manager is an addition to the APL*PLUS TOOLS product line. STSC's APL*PLUS TOOLS series is an extensive library of standardized, consistent, and well-documented programmers' tools and application building software. For detailed descriptions of other popular APL*PLUS TOOLS software products, contact STSC, Inc. Start saving hours of development time with this and other components of the exciting APL*PLUS TOOLS series.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

APL*PLUS is a registered trademark and service mark of STSC, Inc. PLUS*WARE is a trademark of STSC, Inc. LOTUS, 1-2-3, and Symphony are trademarks of Lotus Development Corporation. IBM is a registered trademark of International Business Machines.

Copyright ©1986, STSC, Inc.
Printed in the U.S.A.
CC212-0486

A PLUS★WARE™ PRODUCT

APL★PLUS®
SYSTEM ADD-INS

STSC's APL★PLUS System Add-Ins are *productivity boosters* for your APL★PLUS System for the PC.

Have you ever wished you could use standard APL notation within a Lotus® 1-2-3® spreadsheet? Perform quick ad hoc calculations using proven subroutines? Make more effective use of your PC? Standardize your approach to common software development tasks?

If you answered "Yes" to any of these questions, read on! We've got options that will put you in *overdrive.*

STSC offers the following add-in products to increase your productivity:

- APL★PLUS PC TOOLS
- Spreadsheet Manager for the PC
- IRMA® Interface for the PC
- Financial and Statistical Library

These products will . . .

- allow you to *focus* on solving problems rather than writing utility programs.
- *accelerate* the progress of your application building efforts.
- *expand* the capabilities of your microcomputer applications.
- *standardize* your approach to common application development requirements.

APL★PLUS PC TOOLS

APL★PLUS PC TOOLS is truly the *professional's package,* providing the serious application developer with the *competitive edge.*

A collection of efficient, proven utilities to address common development tasks, which allows you to concentrate on the unique aspects of your problem.

- The Screen Manager
- The Reporting and Output Formatter
- The File Manager
- The Application Maintainer
- The Exception Handler
- The Host

THE SCREEN MANAGER

Control the look and feel of your applications with our powerful screen manager.

This manager contains programs to present selection menus or input forms to handle interactions with and interpret responses of an application user. Maintenance programs allow you to build, edit, maintain, file, and prepare for rapid display screen input form definitions and menu definitions encompassing an extensive range of features. The screen editor allows you to design the text and lines on the screen and the display attributes (colors, highlights, reverse-video, etc.) The editor is used to develop help screens, reference panels, title screens, data input forms and more.

THE REPORTING AND OUTPUT FORMATTER

Command attention with reports that present information in an organized and compact manner.

Our output utilities help you manage multi-page reports, including page numbering, titling, and footnoting. Direct output to screen, printer, or disk file, according to the requirements of your application.

THE FILE MANAGER

Speed up disk operations and *track* files with the Disk and File Manager.

Software to produce formatted file listings, document file layouts, review and edit the components of APL or native files, and audit changes to the files is included. Also, a software device driver is used to reserve a portion of the computer's RAM so that it appears to DOS 2.00 to be an additional disk storage device, thus permitting extremely fast disk operations.

THE APPLICATION MAINTAINER

Advance software development with these proven APL★PLUS System utilities, renowned for increasing the productivity, reliability, and useability of APL applications.

These utilities are a must in building, documenting, debugging, and maintaining applications, and providing error trapping routines that are so essential to developers and users alike.

- Utility functions for searching, analyzing, modifying, and maintaining functions, workspaces, and file images of workspaces
- Utilities to compare workspaces, variables, and functions
- Utilities to capture a file image of a workspace and to re-create a workspace from a file image
- Workspace documentor and cross-reference facility.

THE EXCEPTION HANDLER

Free your mind to *focus* on the application by allowing The Exception Handler to deal with errors.

The Exception Handler includes productivity aids for the programmer and security and recovery aids for the application. With the Exception Handler, all the error analysis, context checking, interrupt control, and stack management are pre-programmed into the handler's utilities.

THE HOST

Broaden your horizon by setting up your PC as a host computer.

You can be anywhere in the world, and still run your APL applications from a remote terminal, even through a dial-up telephone connection. And you can transfer individual variables or entire files between PCs.

SPREADSHEET MANAGER FOR THE PC

Get *added leverage* with the Speadsheet Manager. This is the *businessman's package,* combining the capabilities of APL with spreadsheet capabilities and linking APL with popular software application packages.

The spreadsheet editor allows you to move a collection of APL variables into a spreadsheet. You can insert titles, move or sort rows and columns, execute and store APL formulas for individual cells or ranges of cells, and store any portion of the data in APL variables for easy retrieval.

With just a few simple commands, you can import Lotus or ASCII files into APL variables, or export data in an APL spreadsheet to external sources.

IRMA INTERFACE FOR THE PC

Channel the energy of your PC by using the IRMA interface—the *communicator's package*—to connect with your mainframe.

Use your PC and IRMA Board as a 3278 Model 2 terminal to an IBM® mainframe computer. Transfer APL workspaces and files to and from an IBM mainframe. Control mainframe full-screen applications from your PC. All via your IRMA Board Interface.

FINANCIAL AND STATISTICAL LIBRARY

Maximize your investment by using over 200 prewritten routines provided in the *analyst's package*—the Financial and Statistical Library.

Whether you're performing ad hoc calculations or building complex applications, you'll come to rely on this high-performance collection of financial and statistical programs. Our Financial and Statistical Library includes routines to:

- Perform financial calculations
- Manipulate and forecast time series
- Compute statistical measures
- Generate probability distribution samples.

The library's depth is evident in its 5 different types of loan amortization, 6 types of depreciation, 8 types of rate of return, more than 15 types of forecasting, and more than 30 probability distributions. And most programs accept parameters to customize the calculations even further.

SYSTEM REQUIREMENTS

The APL★PLUS System Add-Ins are designed to operate under the APL★PLUS System for the PC, Version 5.0 or later. DOS Version 2.0, or later and a minimum of 384K RAM is suggested, although many utilities will operate with less. Color monitors are supported but not required.

The Spreadsheet Manager requires a computer that is screen-compatible and keyboard-compatible with the IBM PC. The Spreadsheet Manager will handle Lotus 1-2-3 (Release 1, 1A, and 2), and Symphony,™ (Release 1.0 and 1.1) worksheets.

Use of the IRMA Interface for the PC requires an installed IRMA Board with accompanying software, Version 1.42 or later, not included with our IRMA Interface. IRMA support software also requires GDDM and APL★PLUS, VS APL, or APL2 on the mainframe.

GET THE WINNING ADVANTAGE—ORDER TODAY

To order your APL★PLUS System Add-Ins see your local dealer or contact STSC.

800-592-0050
301-984-5123 (In Maryland)

In Canada or internationally:
301-984-5412
Telex 898085 STSC ROVE
Telefax 301-984-5094

STSC: THE COMPANY

STSC, Inc., located in Rockville, Maryland, is an industry leader in the development of the APL programming language. STSC delivers APL-based software for the microcomputer, minicomputer, and mainframe environments, and publishes microsoftware written by independent developers. STSC licenses software through distributors in several countries worldwide.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
800-592-0050
301-984-5123 (In Maryland)

In Canada or internationally:
301-984-5412
Telex 898085
Telefax 301-984-5094

APL★PLUS and PLUS★WARE are registered trademarks of STSC, Inc. Lotus and 1-2-3, IBM, and IRMA are registered trademarks of Lotus Development Corporation, International Business Machines Corporation, and Digital Communications Associates, Inc., respectively. Symphony is a trademark of Lotus Development Corporation.

Other product names mentioned in this brochure may also be trademarks used here for identification only.

Copyright © 1987, STSC, Inc.
CC200-0887
Printed in the U.S.A.

A PLUS★WARE™ PRODUCT

POCKET APL™

STSC's Pocket APL is the perfect package for you to begin enjoying the benefits of APL. Pocket APL is a complete APL implementation—and more. More than a language processor, features include an online calculator, full-screen capabilities and windowing, report formatting, a comprehensive online HELP facility, dual file systems, an english keyword option, and enhanced error trapping. Over 50 system functions help you quickly solve your problems. And, best of all, as easy as Pocket APL makes programming, it's that easy on your pocket too!

The Pocket APL software is accompanied by introductory texts and learning exercises so that even first-time users will find APL easy to learn and use. Pocket APL workspaces and files are compatible with STSC's APL★PLUS® PC System, allowing an easy conversion of your programs to an advanced environment.

Pocket APL has minimal hardware requirements and operates with as little as 128K of memory. The product runs on the IBM® Personal Computer, the IBM PCjr, and true compatibles equipped with the PC DOS or MS-DOS® operating systems.

SYSTEM FEATURES

- A complete implementation of APL
- A soft character set is provided to display APL symbols on computers with an IBM compatible graphics adapter. An optional character ROM may be purchased separately for monochrome display adapters. English keywords instead of APL symbols are also available.
- Workspaces up to 64K
- Online HELP facility
- Full-screen capabilities and windowing
- Report formatting
- Dual file systems
- Enhanced error trapping
- Over 50 system functions
- Comprehensive documentation.

SYSTEM SPECIFICATIONS AND LIMITS

DATA TYPES

Integer: 16 bits per element
Floating Point: 64-bit IEEE Standard double precision
Character: 8 bits per element

SYSTEM LIMITS

Numeric range: -1.797E308 to 1.797E308
Maximum rank: 63
Maximum variable size: workspace size
Maximum coordinate length: 32,767
Maximum lines in function: 9999
Maximum symbols: 2048
Maximum workspace: 65,534 bytes

SYSTEM FUNCTIONS

APL Component Files:

- ☐ *FAPPEND* —Append array to component file
- ☐ *FCREATE* —Create component file
- ☐ *FDROP* —Drop components from file
- ☐ *FERASE* —Erase component file
- ☐ *FLIB* —List component files in a directory
- ☐ *FNAMES* —List names of tied files
- ☐ *FNUMS* —List tie numbers of tied files
- ☐ *FREAD* —Read component from file into workspace
- ☐ *FREPLACE* —Replace component
- ☐ *FSIZE* —Report file size information
- ☐ *FSTIE* —Share tie (same as ☐ *FTIE*)
- ☐ *FTIE* —Exclusive tie
- ☐ *FUNTIE* —Untie (close) files

Native DOS Files:

- ☐ *NAPPEND* — Append APL data to regular DOS file as byte stream
- ☐ *NCREATE* — Create DOS file
- ☐ *NERASE* — Erase DOS file
- ☐ *NNAMES* — List names of tied (open) DOS files
- ☐ *NNUMS* — List tie numbers of open DOS files
- ☐ *NREAD* — Read data from DOS file into workspace
- ☐ *NREPLACE* — Replace date in DOS file
- ☐ *NSIZE* — Report size of DOS file
- ☐ *NTIE* — Open DOS file
- ☐ *NUNTIE* — Untie (close) DOS files

Saved Workspaces:

- ☐ *LOAD* — Load workspace under program control
- ☐ *QLOAD* — Load workspace (quietly) under program control
- ☐ *WSLIB* — List saved workspaces in a directory
- ☐ *XLOAD* — Load workspace without executing latent expression

Active Workspace Environment:

- ☐ *AI* — Accounting Information
- ☐ *CR* — Canonical representation of a function
- ☐ *CT* — Comparison tolerance
- ☐ *DEF* — Define a function from character form
- ☐ *DL* — Delay for specific interval
- ☐ *DR* — Report or change internal data type for variable
- ☐ *EX* — Erase a function or variable
- ☐ *FX* — Define a function from canonical form
- ☐ *IO* — Index origin
- ☐ *LC* — Line counter
- ☐ *LX* — Latent expression
- ☐ *NC* — Name class of an APL object
- ☐ *NL* — List objects of a given class in workspace
- ☐ *PP* — Print precision
- ☐ *PW* — Print width
- ☐ *RL* — Random number seed
- ☐ *SI* — Report current state indicator stack
- ☐ *SINL* — Display state indicator and localized names
- ☐ *SIZE* — Report size of an APL object
- ☐ *STOP* — Report or set stop vector on a function
- ☐ *SYMB* — Report current symbol table size and number used
- ☐ *TRACE* — Trace function execution
- ☐ *VR* — Visual representation of function
- ☐ *WA* — Amount of unused workspace area available
- ☐ *WSID* — Name of active workspace

Exception Handling:

- ☐ *ALX* — Program control of attention interrupts
- ☐ *DM* — Most recently produced error message
- ☐ *ELX* — Program control of APL errors
- ☐ *ERROR* — Signal user-defined error under program control

Formatting:

□*FI*	—Input conversion of character to numeric
□*FMT*	—General purpose number and text formatter
□*VI*	—Validate numeric input

Communications and Peripherals:

□*CURSOR*	—Move cursor to position on screen or reports current position
□*INKEY*	—Read first character typed at keyboard
□*WGET*	—Get data from screen
□*WINDOW*	—Read or set current window
□*WPUT*	—Write data to screen

Interface to DOS and non-APL Programs:

□*LIB*	—Report all files in a directory, regardless of type
□*PEEK*	—Retrieve value of memory location
□*POKE*	—Replace memory location with new value

Miscellaneous:

□*AV*	—Atomic vector (all possible characters)
□*HELP*	—Call HELP facility
□*KEYW*	—Set or retrieve keyword state
□*SS*	—String search function
□*SYSID*	—APL★PLUS System identification
□*TCBEL*	—Terminal control constants (also □*TCBS*, □*TCDEL*, □*TCESC*, □*TCFF*, □*TCLF*, □*TCNL*, □*TCNUL*)
□*TS*	—Current time stamp

System Commands:

)	—Recall last APL statement for editing
)*CLEAR*	—Clear workspace
)*COPY*	—Copy from saved workspace
)*DROP*	—Delete saved workspace
)*ERASE*	—Erase APL objects
)*FLIB*	—List component files in a directory
)*FNS*	—List functions in workspace
)*HELP*	—Activate HELP facility
)*KEYWORD*	—Change or display keyword state
)*LIB*	—List all files in a directory
)*LOAD*	—Load a saved workspace
)*OFF*	—End APL session
)*PCOPY*	—Copy from saved workspace without replacement
)*RESET*	—Clear state indicator
)*SAVE*	—Save the active workspace
)*SI*	—Display state indicator
)*SIC*	—Clear state indicator
)*SINL*	—Display state indicator and localized names
)*SYMBOLS*	—Display or change size of symbol table
)*VARS*	—List variables in a workspace
)*WSID*	—Display or change name of active workspace
)*WSLIB*	—List workspaces in a directory
)*XLOAD*	—Load workspace suppressing execution of latent expression

PRODUCT CONTENTS

The Pocket APL software package comes complete with award winning documentation. The package includes:

- **APL Is Easy!** An introductory text that takes you through the simple steps of developing your first applications in APL.
- Comprehensive reference guide. Filled with useful summaries of APL language features, system features and functions.
- Keyboard reference card. For easy cross-reference between APL symbols and keywords.
- APL keyboard placard. Quick, at-a-glance reference for the keyboard positions of APL symbols.
- System software and workspace disk. Everything you need to turn your microcomputer into a fast, powerful problem-solver.

AVAILABILITY

STSC's Pocket APL software is available through local dealers or by contacting STSC, Inc.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

APL★PLUS is a service mark and trademark of STSC, Inc.
Pocket APL and PLUS★WARE are trademarks of STSC, Inc.
IBM and MS-DOS are registered trademarks of International Business Machines and Microsoft, respectively.

Copyright © 1986, STSC, Inc.
Printed in the U.S.A.
CC197-0686

A PLUS★WARE™ PRODUCT

APL★PLUS® II System for the 80386

STSC's APL★PLUS II System for the 80386 is the new generation of the powerful APL programming language. Designed to take full advantage of 80386 technology, APL★PLUS II combines the best features of the APL★PLUS Systems for mainframes and microcomputers. And, because APL★PLUS II is **fast**, your applications run in record time.

Built-in performance-oriented features help you work efficiently. APL★PLUS II's unique combination of features includes nested arrays, pull-down menus, a powerful full-screen ring editor, and workspace size limited only by the amount of RAM in your computer. You can even redefine the keyboard and set up function keys to suit your particular needs, and store and access data in DOS files outside your APL workspace.

APL★PLUS II also has extensive graphics capabilities, including APL graphics system functions and industry standard VDI graphics and GSS*CGI interface.

SYSTEM REQUIREMENTS

PC or PS/2 compatible computer with 80386 CPU
80387 or 80287 coprocessor strongly recommended
2 MB RAM, 4 MB recommended for a 3 MB workspace
640K DOS memory
DOS, Version 3.3 or later
One high-density floppy disk; hard disk strongly recommended

DATATYPES

Boolean:	1 bit per element
Integer:	32 bits per element
Floating point:	64-bit IEEE Standard double precision
Character:	8 bits per element
Nested:	32-bit pointer element
Heterogeneous:	mixed datatypes

SYSTEM LIMITS

Numeric limits:	-1.797E308 to 1.797E308
Maximum rank:	127
Maximum variable size:	size of workspace
Maximum coordinate length:	2,147,483,647
Maximum lines in function:	9999
Maximum workspace symbols:	32767
Maximum identifier length:	100 characters
Maximum workspace size:	System limit: 2,147,483,640 bytes Practical limit: available RAM

SYSTEM FEATURES

- A complete APL★PLUS II implementation in an extended working environment
- Workspace size specified by user and limited only by available RAM
- Object size limited only by workspace size
- Nested arrays compatible with STSC's mainframe, VAX, and UNIX APL★PLUS Systems, generally compatible with IBM's APL2
- Uses the Phar Lap 386|DOS-Extender to access the DOS environment from protected mode
- Two complete interpreters: one designed to take full advantage of an installed math coprocessor
- Custom configuration file sets user preferences automatically at APL startup
- Use of pull-down menus or keystroke combinations to set and change system parameters
- Traditional APL or ASCII-like Text (unified) keyboard
- Table-driven keyboard driver completely customizable; supports international keyboard layouts
- Ability to switch between APL and DOS keyboard drivers
- Programmable function keys and ability to treat each key as a function key
- Full-screen ring editor, allowing simultaneous editing of APL objects, native files, and the APL session itself
- User-specified memory allocation for off-screen scrolling allows retention of thousands of lines of session log
- Ability to specify files and workspaces by library number or by DOS directory path
- APL★PLUS PC ⎕G-style graphics functions and new VDI graphics capabilities
- Ability to execute DOS commands directly
- Component file system
- Interfaces to native DOS files
- Facility for reading and writing APL★PLUS PC files
- Local area network support for concurrent sharing of files and workspaces
- Report-formatting facility
- Error-trapping features
- Partial compilation avoids overhead of syntax analysis on re-executed statements
- Dynamic symbol table enlarges as required
- Over 100 system functions and system commands
- A suite of utility workspaces
- ⎕AV completely compatible with APL★PLUS PC
- Terminal emulation and upload/download capability
- Source level transfer capability

⎕*DL*	Delay for a specific interval
⎕*DR*	Report internal datatype of variable
⎕*ERASE*	Erase a function or variable
⎕*EX*	Erase a function or variable
⎕*FX*	Define a function from canonical form
⎕*IDLIST*	List objects in workspace
⎕*IDLOC*	Report definitions of localized identifiers
⎕*IO*	Index origin
⎕*LC*	Line counter
⎕*LOCK*	Lock a function under program control
⎕*LX*	Latent expression
⎕*MF*	Monitor function executions
⎕*NC*	Name class of an APL object
⎕*NL*	List of workspace object names
⎕*PP*	Print precision
⎕*PR*	Prompt replacement for ⍞ input/output
⎕*PW*	Print width
⎕*RL*	Random number seed
⎕*SI*	Report current state indicator stack
⎕*SINL*	Display state indicator and localized names
⎕*SIZE*	Report size of an APL object
⎕*STOP*	Report or set a stop vector for a function
⎕*SYMB*	Report current symbol table size
⎕*TRACE*	Trace function execution
⎕*VR*	Visual representation of a function
⎕*WA*	Amount of unused workspace available
⎕*WSID*	Name of the active workspace
⎕*WSSIZE*	Size of current workspace
⎕*WSOWNER*	Identify owner of active workspace
⎕*WSTS*	Timestamp of current active workspace

Exception Handling

⎕*ALX*	Latent expression handler for weak interrupts
⎕*DM*	Most recently produced error message
⎕*ELX*	Latent expression handler for APL errors
⎕*ERROR*	Signal APL error under program control
⎕*SA*	Stop action: exit function, clear stack, exit APL, or signal error upon return to immediate execution

Formatting

⎕*FI*	Convert character to numeric input
⎕*FMT*	General purpose report formatter
⎕*VI*	Verify input conversion

Communications and Peripherals

⎕*ARBIN*	General purpose input/output function for communicating with peripherals and remote computers. Handles port selection, translation, overstriking, timeouts, buffer sizes, and terminator characters
⎕*ARBOUT*	Transmit arbitrary character sequences to screen
⎕*CRT*	Report current state of screen memory
⎕*CURSOR*	Move cursor to position on screen or report current position
⎕*INBUF*	Insert keystrokes into keyboard event buffer
⎕*INKEY*	Read a single keystroke from the buffer
⎕*PFKEY*	Read or set function keys
⎕*WGET*	Get data from screen
⎕*WIN*	Multi-window screen input processor
⎕*WINDOW*	Read or set current active window
⎕*WKEY*	Define keystroke behavior for ⎕*WIN*
⎕*WPUT*	Write data to screen

Interface to DOS and Non-APL Programs

⎕*CALL*	Call Assembler routine in APL workspace
⎕*CHDIR*	Report or change current working directory
⎕*CMD*	Execute DOS commands under program control
⎕*INT*	Call a system interrupt
⎕*LIBD*	Assign library number to DOS directory
⎕*LIBS*	List library number to directory assignment
⎕*MKDIR*	Create DOS directory
⎕*PEEK*	Retrieve value of memory location
⎕*POKE*	Replace memory location with new value
⎕*RMDIR*	Remove DOS directory
⎕*SEG*	Set or retrieve segment register value
⎕*STPTR*	Locate memory address of an APL object

Graphics Functions

⎕*GCIRCLE*	Draw circles, arcs, or ellipses
⎕*GINIT*	Initialize graphics adapters
⎕*GLINE*	Draw lines, points, and pixels
⎕*GMASK*	Specify mask pixels when drawing lines or circles
⎕*GPAINT*	Fill in region of screen in specified color
⎕*GPRINT*	Copy graphics image to printer
⎕*GSHADE*	Shade rectangles with color or patterns
⎕*GSTATUS*	Report on graphics display parameters
⎕*GTYPE*	Select or report type of pixel interaction when drawing new lines
⎕*GVIEW*	Select or report the area of the screen to project graphics image
⎕*GWINDOW*	Select or report the portion of the graphics easel to display on the screen
⎕*GWRITE*	Write characters on graphics screen
⎕*GZOOM*	Set magnification factor for graphics characters

Miscellaneous

⎕*AV*	Atomic vector
⎕*ED*	Use full-screen editor under program control
⎕*EDIT*	Edit object in full-screen editor

- Interfaces to non-APL programs and packages
- User-definable Help facility
- Ability to recall previous lines for editing and re-execution while maintaining APL session integrity
- Workspaces for calculations on complex numbers and for finding eigenvalues and eigenvectors
- Extensive documentation

ENHANCED LANGUAGE FEATURES

Nested arrays
- Ability to enclose an array to form a scalar that can be an element of another array with no limit on depth of nesting

Heterogeneous arrays
- Ability to mix character and numeric data in same array

Extended operators
- Reduction, scan, inner product, and outer product work with all primitive, system, user-defined, and derived functions
- New "each" operator applies any function to each element of an array

System Extensions for Nested Arrays
- All scalar, selection, and structural primitive functions work with nested and heterogeneous arrays
- Complete support for nested arrays in component files
- Attractive display and default formatting of nested arrays

Enhanced Primitive Functions

l ⊂ *r*	**Find,** pattern match on subarrays
l ~ *r*	**Without,** set difference
⊂ *r*	**Enclose,** create nested array
l ⊂ *r*	**Partitioned enclose,** add nesting
↓ *r*	**Split,** enclose one coordinate
↑ *r*	**Mix,** disclose and raise rank
⊃ *r*	**Disclose,** remove level of nesting
l ⊃ *r*	**Pick,** select from nested array
∈ *r*	**Type,** prototypical value
≡ *r*	**Depth,** level of nesting
l ≡ *r*	**Match,** equivalence of two arrays
r[⊂*l*]	**Choose (scatter-point) indexing**
r[⊂*l*]←*a*	**Choose assignment**
r←(*a*)(*b*)(*c*)	**Strand notation**
a b c ← *r*	**Strand assignment**

SYSTEM FUNCTIONS

APL Component Files

⎕*FAPPEND*	Append array to component file
⎕*FAVAIL*	Report file system availability
⎕*FCREATE*	Create component file
⎕*FDROP*	Drop components from file
⎕*FDUP*	Duplicate file and compact unused space
⎕*FERASE*	Erase component file
⎕*FHIST*	Display historical timestamps for file
⎕*FHOLD*	Hold file, synchronization by queue
⎕*FLIB*	List component files in directory
⎕*FNAMES*	List names of tied files
⎕*FNUMS*	List tie numbers of tied files
⎕*FRDAC*	Read file access matrix
⎕*FRDCI*	Read audit trail information on component
⎕*FREAD*	Read component from file into workspace
⎕*FRENAME*	Rename component file
⎕*FREPLACE*	Replace a component with a new value
⎕*FRESIZE*	Change size limit on file
⎕*FSIZE*	Report file size information
⎕*FSTAC*	Set file access matrix
⎕*FSTIE*	Share tie (open for concurrent access)
⎕*FTIE*	Exclusive tie (nonconcurrent access)
⎕*FUNTIE*	Untie (close) files

Native DOS Files

⎕*LIB*	List names of files in a directory
⎕*NAPPEND*	Append APL data to native DOS file
⎕*NCREATE*	Create DOS file
⎕*NERASE*	Erase DOS file
⎕*NNAMES*	List names of tied (open) DOS files
⎕*NNUMS*	List tie numbers of tied (open) DOS files
⎕*NREAD*	Read data from file
⎕*NRENAME*	Rename file
⎕*NREPLACE*	Replace data in file
⎕*NSIZE*	Report size of file
⎕*NTIE*	Tie (open) native file
⎕*NUNTIE*	Untie (close) native file

Saved Workspaces

⎕*COPY*	Copy objects under program control
⎕*LOAD*	Load workspace under program control
⎕*PCOPY*	Protective copy under program control
⎕*PSAVE*	Protective workspace save
⎕*QLOAD*	Quiet load under program control
⎕*SAVE*	Save workspace under program control
⎕*WSID*	List saved workspaces in a directory
⎕*XLOAD*	Load workspace without executing ⎕*LX*

Active Workspace Environment

⎕*AI*	Display accounting information
⎕*CR*	Canonical representation of a function
⎕*CRL*	Canonical representation of function line
⎕*CRLPC*	Extract public comment from a function
⎕*CT*	Comparison tolerance
⎕*DEF*	Define a function from visual form
⎕*DEFL*	Replace/insert a line in a function

□*HELP*	Call Help facility
□*SOUND*	Play music on system speaker
□*SS*	String search
□*SYSID*	APL★PLUS II System identification
□*SYSVER*	APL★PLUS II System version number
□*TC*	Terminal control character combination (backspace, linefeed, newline)
□*TCxx*	Terminal control constants (□*TCBEL*, □*TCBS*, □*TCDEL*, □*TCESC*, □*TCFF*, □*TCHT*, □*TCLF*, □*TCNL*, □*TCNUL*)
□*UL*	Number of users on the system; always 1 in APL★PLUS II
□*USERID*	Report network ID of machine

SYSTEM COMMANDS

)	Recall last APL statement
)*CMD*	Execute DOS command
)*CLEAR*	Clear the workspace, optionally change size
)*COPY*	Copy from saved workspace
)*DROP*	Delete saved workspace
)*EDIT*	Create edit session and edit named object
)*ERASE*	Erase APL objects
)*FLIB*	List component files in a directory
)*FNS*	List all functions in a workspace
)*HELP*	Invoke Help facility
)*LIB*	List all files in a directory
)*LIBS*	Display library-directory correspondence
)*LOAD*	Load a saved workspace
)*OFF*	End an APL session
)*PCOPY*	Protected copy from saved workspace
)*PSAVE*	Protected save of a workspace
)*RESET*	Clear state indicator
)*SAVE*	Save the active workspace
)*SI*	Display contents of state indicator stack
)*SIC*	Clear state indicator
)*SINL*	Display state indicator with names localized
)*SYMBOLS*	Display or change the size of the symbol table
)*VARS*	List all variables in a workspace
)*WSID*	Display or change the name of the active workspace
)*WSLIB*	List the workspaces in a directory
)*XLOAD*	Load a workspace without executing □*LX*

PRODUCT CONTENTS

The APL★PLUS II System consists of:

- Three high-density disks (5.25″ or 3.5″) containing system software, APL soft character sets, utilities, and GSS drivers (low-density disks available on request)
- Installation instructions
- APL keyboard labels
- Complete documentation, including User's Guide, Reference Manual, GSS*CGI Programmer's Guide, and Quick Reference Guide
- ROM request form (one character ROM available free of charge)
- Additional GSS*CGI drivers available at extra cost

PORTABILITY

APL★PLUS II provides an extremely high level of upward compatibility with the APL★PLUS PC System for MS-DOS. APL★PLUS II is generally compatible with STSC's other APL★PLUS Systems, including those running in the VM/CMS, MVS/TSO, VAX under VMS, UNIX, and Macintosh environments. Built-in communications and source level transfer software makes it easy to transfer data and programs to and from various computer systems.

AVAILABILITY

APL★PLUS II is available through STSC in the U.S. or local dealers elsewhere. A run-time version of the software is available for application developers with APL-based software products.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
U.S.A.
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)
Telex #898085 STSC ROVE
Telefax (301) 984-5094

APL★PLUS is a registered trademark and service mark of STSC, Inc.
IBM is a registered trademark of International Business Machines.
VAX and VMS are trademarks of Digital Equipment Corporation.
UNIX is a trademark of AT&T Bell Laboratories.
Macintosh is a trademark licensed to Apple Computer, Inc.
GSS*CGI is a trademark of Graphic Software Systems, Inc.
386 | DOS-Extender is a trademark of Phar Lap Software, Inc.

Copyright © 1988 STSC, Inc.
Printed in the U.S.A. 06885M

ENHANCEMENTS SUMMARY FOR VERSION 2.0
APL★PLUS® II FOR THE 80386

STSC's newest release of APL★PLUS II sets new standards of performance, power, and ease of use. Designed with advice from enthusiastic APL users, the collection of enhancements in Version 2.0 provides major improvements to your productivity and efficiency.

Interface Non-APL Programs with ⎕NA

This new feature allows you to call compiled subroutines from APL. Such routines often perform iterative calculations faster than is possible with APL primitives. ⎕NA also opens the door from APL to a wealth of software available for the PC—database managers, graphics packages, and scientific and statistical subroutine libraries.

One of the difficulties in working with compiled languages is building interactive user interfaces. ⎕NA solves this problem. You simply use APL as the interactive user interface. Then it's as easy to call a C routine as it is to call a defined APL function!

Interfacing is simple with ⎕NA. You write, compile, and link non-APL routines in the usual way, then use ⎕MLOAD to load them into the APL workspace. ⎕NA is used to describe each routine's parameters to APL. The result is an APL function—an "associated function"—that forms the interface to the routine.

Subroutine calls by way of associated functions are efficient and versatile. You don't have to worry about datatypes—APL automatically converts data to the type required by the non-APL routine.

Version 2.0 includes interfaces for several compilers. For the full power of the 80386, you can use the MetaWare High-C compiler to write routines that run in the 32-bit protected mode. Or you can use popular industry-standard C compilers like Microsoft C and Borland's Turbo C. Separate, installable interface programs are used to bridge from APL to each compiler. Source code for these interfaces is included, allowing advanced users to build interfaces to other languages.

Interpreter Performance Improvements

Speedups to the APL★PLUS II System make its superb performance even better. Formatting is quicker and the interpreter delivers faster and more consistent performance in large and complex applications.

Full-Screen Editing of Numeric Arrays

Using the enhanced full-screen editor in Version 2.0, you can edit numeric matrices just as easily as you now edit functions or character arrays. Numeric editing is fully integrated into the session manager. You can search and replace, cut and paste, use session-manager menus, pull in arrays from the workspace, or select subarrays to save into variables.

Numeric editing promotes the usefulness of APL for casual, hands-on data analysis. If you have ever wished for an easy facility for the input and editing of numeric data, you will love the numeric editor.

Other Session Manager Enhancements

The APL★PLUS II session manager is the most productive APL environment ever designed. New enhancements make it even more productive.

- "Point-and-Edit"

  The Point-and-Edit feature makes it easy to trace program logic through subroutine calls when debugging. To use Point-and-Edit, you place the cursor on the name of the object that you want to edit. A single keystroke suspends the current editing session and begins editing the object. You don't even have to type the object's name!

- Customize the Terminal Emulator

  The enhanced KEYBOARD workspace can be used to define the precise local behavior and transmitted codes for every possible keystroke in terminal mode.

- Built-in Print Spooler

  This feature lets you work while the session manager directs other output to the printer.

- Customized PF Keys

  With Version 2.0 you can define PF keys in the configuration file. This allows you to define permanent PF key definitions that remain from session to session.

Other Valuable Tools and Utilities

- CONVERT Workspace

  This workspace automates the conversion from APL★PLUS PC to APL★PLUS II and back again. As it works, CONVERT scans APL functions and flags lines where revisions may be needed. CONVERT includes emulators for ⎕IN and the QPACK workspace of APL★PLUS PC.

- FFT Workspace

  The FFT workspace contains APL utilities to generate Fast Fourier Transforms and their inverses.

- EIGENVAL Workspace

  EIGENVAL has been enhanced to calculate complex eigenvectors.

- HANDLERS Workspace

  This workspace contains an easy, consistent, self-documenting technique for incorporating reliable error-and-interrupt handling into your applications. This workspace is compatible with the HANDLERS workspace sold as an add-on to APL★PLUS PC.

- CODEPAGE

  The CODEPAGE workspace lets you produce APL characters using the Code Page facilities introduced in DOS Version 3.30. This workspace also contains functions that allow you to edit character fonts for EGA and VGA displays.

- VDI

  New utilities are included for quick and easy line plots. Scaling and labeling are done automatically.

Virtual Memory Support

APL★PLUS II is now capable of running under the Phar Lap Virtual Memory Manager (available from Phar Lap and various dealers). With virtual memory you can use larger workspaces than would be possible with installed RAM, although performance may be affected.

APL Language Enhancements

Derived functions using "each" and "outer product" on non-scalar functions now accept empty arguments instead of producing NONCE ERROR. APL★PLUS II adapts the definitions used in IBM's mainframe APL2 system for this behavior.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland)

(301) 984-5412 (Outside the U.S.
or contact your local dealer)

Telex #898085 STSC ROVE
Telefax (301) 984-5094

APL★PLUS is a registered trademark of STSC, Inc.
Other product names mentioned in this brochure may also be trademarks used here for identification only.

Copyright © 1989, STSC, Inc.
CC237-0689
Printed in the U.S.A.

VERSION 3 ENHANCEMENTS SUMMARY

APL★PLUS® II SYSTEM FOR THE 80386

APL★PLUS II, Version 3, STSC's latest release for 80386 computers, offers a host of new features to boost productivity and programming efficiency, without sacrificing any of the speed, flexibility, or capacity that you have come to rely on. It's a step forward in productivity with less effort.

Mouse Support

Topping the list of Version 3's new user-friendly enhancements is support for the IBM® Personal System/2™ Mouse and the Microsoft™ Mouse, which makes editing in the session manager quick and convenient. You now have point-and-click access for selecting from menus; moving around the edit ring; scrolling through sessions; and tagging, copying, and moving blocks of text or numbers — within a session or among different sessions. Mouse support can be incorporated into your applications as well.

New Interfaces

- Lotus® 1-2-3® — Data can be easily imported from a Lotus 1-2-3 worksheet for complex calculations using APL★PLUS II, then exported back to Lotus. Moving data in and out of Lotus is simple, and you won't have to re-enter data or write complicated Lotus programs to derive the right data.
- IRMA® — Channel the energy of your computer by using the IRMA communications interface to transfer APL workspaces and files to and from an IBM mainframe. Control mainframe full-screen applications from your 80386 — all via your IRMA interface.
- ⎕*NA*/FORTRAN — The ⎕*NA* interface opens the door from APL to FORTRAN. The interface includes support for programs compiled in real mode with Microsoft FORTRAN, Version 5.0, and programs compiled in protected mode with MicroWay's NDP FORTRAN-386,™ Version 2.0.

Productivity Enhancements

- User Command Processor — Version 3 includes a user command processor (integrated into the interpreter) that lets you use utilities as you use system commands. You can define and store a set of commands that are always available, regardless of the workspace you are using. More than 25 predefined utility commands are included in Version 3, so you can easily manipulate objects in the active workspace or in command files.
- WSDOC Workspace — The Workspace Documentation Facility generates a variety of reports about the contents of a workspace.
- OUTPUT Workspace — This utility helps you manage multi-page reports, including page numbering, titling, and footnoting. And you can direct your output to screen, printer, or disk file, according to the requirements of your application.
- Other Utility Workspaces — Additions to the WSCOMP, FILEDOC, WFCARE, and UTILITY workspaces help you compare and maintain functions, files, and workspaces.

Additional DOS Memory

Version 3 provides access to additional DOS memory. The session manager and the interpreter have been merged in extended memory, and both run in protected mode, freeing up DOS memory for your use. You can now load more terminate-and-stay-resident (TSR) programs, such as networking software, without interfering with APL★PLUS II. The new 32-bit session manager also allows larger amounts of memory to be used for editing.

Enhanced Printer and Video Display Support

- Printer Support — Version 3 supports PostScript® printers so you can print APL characters and customize your APL output with a variety of PostScript options. Additional support is also included for non-PostScript printers.

- Color Support — The latest release makes it easier to take advantage of your color monitor. You can use session startup parameters to select different colors for the APL session and status line, edit session and status line, partially and fully tagged regions, and the wrapped-line marker. Also new in Version 3 is support for EGA/VGA monitors that allows you to use light background colors instead of the blinking attribute. The new COLOR workspace contains functions that allow you to support color in your applications as well.
- Video Mode and APL Font Support — APL★PLUS II now works with video modes that display more than the standard 80-column, 25-row screen. Display resolutions of 80×43, 132×43, and others are possible, depending on the capabilities of your display adapter. A single program to load and unload APL fonts on EGA and VGA displays also supports the non-default video modes.

System Speedups

- Faster screen interactions through `⎕WIN`, `⎕WPUT`, and `⎕WGET` because the session manager and the interpreter are now integrated.
- Size and file operations are faster on nested arrays.
- Copying objects and workspaces now takes less time with `⎕COPY`, `⎕PCOPY`, `)COPY`, and `⎕PCOPY`.
- The ASMFNS workspace in Version 3 contains some new and revised utility functions that run even more quickly than before.

New Documentation

- Completely reformatted User and Reference Manuals.
- New and easy-to-use Utilities Manual.
- Updated and revised Installation Instructions and Quick Reference Guide.
- New GSS Manual for Version 2.16 drivers.

System Requirements

- PC or PS/2 compatible computer with 80386 or 80486 CPU
- 80387 or 80287 coprocessor strongly recommended
- 2MB RAM, 4MB recommended for a 3MB workspace
- 640K DOS memory
- DOS Version 3.3 or higher
- One high-density floppy disk; hard disk strongly recommended
- IRMA board with terminal emulator to use IRMA interface
- Lotus 1-2-3 Version 1A, 2.0, or 2.2 to use Lotus interface
- IBM Personal System/2™ Mouse or Microsoft™ Mouse to use mouse support

STSC

APL★PLUS is a registered trademark of STSC, Inc.
Other product names mentioned in this brochure may also be trademarks used here for identification only.

Copyright © 1990, STSC, Inc.
CC258-0690
Printed in the U.S.A.

started with APL★PLUS PC. Also included is software that automatically installs the APL★PLUS PC System on your personal computer—making backup disks, creating work files on your hard disk or floppies, and creating an AUTOEXEC.BAT file to automatically start APL when your computer is turned on.

RELEASE 5.0 CONTENTS

The Release 5.0 update package includes:

- Three double-sided diskettes, containing the new APL language processor, HELP files, and completely revised program utilities.
- New clear plastic keyboard stickers, for both standard and unified keyboards.
- Revised placard with the APL character set, also for standard and unified keyboards.
- Quick-start installation guide.
- Update pages documenting the many new features of Release 5.0.
- A handy pocket-size planning calendar and PLUS★WARE™ product catalog.

RELEASE 5.0 SYSTEM REQUIREMENTS

To use Release 5.0 of the APL★PLUS PC System, you must have the following minimum hardware configuration:

- IBM PC or compatible microcomputer with monochrome, color, or EGA display adapter
- 256K RAM memory
- One double-sided disk drive
- DOS 2.0 or later.

STSC
A CONTEL COMPANY

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852

301-984-5123

APL★PLUS and IBM are registered trademarks of STSC, Inc., and International Business Machines, respectively.
PLUS★WARE is a trademark of STSC, Inc.

A PLUS★WARE™ PRODUCT

CC199-1085

RELEASE 5.0 NEW FEATURES SUMMARY

APL★PLUS® PC System

The APL★PLUS PC System, Release 5.0, is the most dramatic development effort that we have undertaken since the initial release of APL★PLUS PC in 1982. The principal design objective of our APL★PLUS Systems has always been to provide the very best application development systems for software developers. Release 5.0 is the best yet!

Release 5.0 represents advancements to the state-of-the-art APL system for microcomputers with new features in four major areas:

- full-screen input & editing
- graphics
- system speedups
- productivity improvements.

FULL-SCREEN EDITING

Numeric Editor

A full-screen editor for numeric arrays, called NEDIT, provides cell-by-cell or extended range full-screen entry and editing of numbers in a matrix. The numeric editor has much of the speed and convenience of a spreadsheet. With NEDIT, you can "compute" the value of a cell or block of cells by entering APL expressions! NEDIT can be called under program control, and provides program hooks, so you can customize the editor to your application!

Full-Screen Window Input

If you write programs that prompt the user for input, you'll find a new system function called ⎕WIN indispensable! This function provides very fast full-screen forms-style input. You simply provide the screen locations (⎕WINDOW definitions) of each field to which you want to enter input, and ⎕WIN will instantly highlight those fields, allowing you to enter data and move between fields. You can optionally customize the default cursor movement definitions, field highlight attributes, and restricted or unrestricted fields; indicate whether input must be numeric or character; and perform customized field validation. Other options, such as auto-tabbing, are also included. The revised INPUT workspace contains all new utility pro-

grams for full-screen data collection, based on ⎕WIN, and is complete with examples.

Full-Screen Editor

The full-screen editor, ⎕EDIT, has been enhanced and is now ⎕WINDOW sensitive. You can restrict the number of lines on the screen used by the editor, so data or errors can remain visible on the screen while they are being adjusted or corrected, or display instructions to a user in one set of rows and use the editor for input in another. New ⎕POKEs are available to tell if a change was made in an edit session, how the session ended, and if a session is pending. Also, new editor commands have been added to perform an intermediate save without leaving the editor and to undo the most recent line deletion. The editor status line now includes information about the keyboard states.

Keyboard Buffer Input

A new system function, ⎕INBUF, allows you to write to the keyboard type-ahead buffer. With ⎕INBUF you can pre-set responses to program prompts, pass data to a workspace being loaded, call the full-screen editor to search through and display data,—the possibilities are endless!

GRAPHICS

Tutorial Workspace

The GRAFDEMO workspace has been completely rewritten to provide a full-screen, menu-driven tutorial of the graphics primitives. The workspace also contains APL functions to produce bar charts, line plots, and pie charts, and contains many other graphics utility functions you'll find immensely useful in building your own graphics applications.

New Graphic Adapters

Release 5.0 supports several new graphic display adapters including the IBM® Enhanced Graphics Adapter (EGA), WANG Professional Computer, and high-resolution support for the AT&T 6300 personal computer.

SYSTEM SPEEDUPS

Assembler Utilities

A new workspace, ASMFNS, has been added, which contains over two dozen commonly needed utility programs, written in Assembler to provide extremely fast execution. These programs can be copied directly into your workspaces to provide dramatic speedups in your applications.

PRODUCTIVITY IMPROVEMENTS

APL 2 Features

Two new language features, "find" (represented by $\sim$) and "without" (represented by $\underline{\in}$) extend the compatibility of APL★PLUS PC with APL2. "Find" extends the string searching ability of ⎕SS to matrices and numeric arrays, and "without" compares the elements of two arrays and returns all of the elements of one array that do not appear in the other. And, the addition of ⎕TC assists conversion efforts from many different mainframe APL systems.

Status Line

A status line has been added that displays information about the APL session. Information contained in the status line includes the keyboard state (e.g., Caps-Lock, insert/replace, numeric pad, etc.) and the status of the APL session (index origin, terminal mode, suspended function, etc.). The status line can be toggled on and off using the Scroll-Lock key.

Unified Keyboard

An optional new "Unified" APL keyboard definition has been added that provides a keyboard layout more consistent with other DOS applications. For example, the right parenthesis key is redefined to the location where it visually appears on the keyboard (shift zero). The Unified keyboard definition makes it easier to move between APL and non-APL applications, and improves the ease of learning for new users. The keyboard definition used in previous releases of APL★PLUS PC is still retained as the default in Release 5.0.

System Overview

A new workspace, APLMENU, provides a full-screen menu-driven overview of commonly-used and advanced features of the APL★PLUS PC System. Topics include graphics, communications, editing, full-screen input, and customizing the APL session environment (e.g., screen color, function keys, default disk drive, etc.). The APLMENU workspace provides many examples of how the current and new features of APL★PLUS PC can be used for developing applications, and contains a number of useful utility programs.

Quick-Start Installation

A brief APL★PLUS PC installation guide provides a quick and easy way for new users to get

APL★PLUS® PC System

Version 5

STSC's APL★PLUS PC System is the most advanced application development system available for the microcomputer. It gives problem-solvers the power to deliver totally integrated, highly efficient solutions to problems that conventional software simply can't handle.

Because of its efficiency, you can develop programs four to ten times faster than with conventional languages. The APL★PLUS PC System can be integrated with other applications you already have, or used to link those that aren't "speaking" to each other.

SYSTEM FEATURES

- A complete APL★PLUS implementation in an extended working environment
- Traditional APL character set or English-like keywords
- Workspaces up to 500K bytes
- Math Coprocessor Support (if installed)
- On-line HELP facility
- High resolution graphics supporting many popular graphics cards, including:
  - — IBM Enhanced Graphics Adapter
  - — IBM Color Adapter
  - — Hercules Graphics Adapters
  - — TECMAR Graphics Master
- Built-in terminal emulation and Upload/Download communications capability
- Full-screen word processing style editor for functions or character data
- Spreadsheet-like editor for numeric data
- Multi-window input manager
- Off-screen scrolling
- User definable function keys
- Programmable keyboard buffer
- Status line
- Interfaces to non-APL programs and packages
- Ability to execute DOS commands directly
- Component file system and interfaces to native DOS files
- Report-formatting facility
- Error-trapping features
- Over 100 system functions and over 200 APL and assembler utility functions
- Extensive documentation.

SYSTEM SPECIFICATIONS AND LIMITS

DATA TYPES

Integer: 16 bits per element
Floating Point: 64-bit IEEE Standard double precision
Character: 8 bits per element

SYSTEM LIMITS

Numeric range: −1.797E308 to 1.797E308
Maximum rank: 63
Maximum variable size: 65,534 bytes
Maximum coordinate length: 32,767
Maximum lines in function: 9999
Maximum symbols: 2048

SYSTEM FUNCTIONS

APL Component Files:

- □ *FAPPEND* — Append array to component file
- □ *FAVAIL* — Report file system availability (not activated)
- □ *FCREATE* — Create component file
- □ *FDROP* — Drop components from file
- □ *FDUP* — Duplicate file and compact unused space
- □ *FERASE* — Erase component file
- □ *FHOLD* — Hold File (not activated)
- □ *FLIB* — List component files in a directory
- □ *FNAMES* — List names of tied files
- □ *FNUMS* — List tie numbers of tied files
- □ *FRDAC* — Read file access matrix
- □ *FRDCI* — Read audit trail information on file component
- □ *FREAD* — Read component from file into workspace
- □ *FRENAME* — Rename component file
- □ *FREPLACE* — Replace component
- □ *FRESIZE* — Change size limit on file
- □ *FSIZE* — Report file size information
- □ *FSTAC* — Set file access matrix
- □ *FSTIE* — Share tie (same as □ *FTIE*)
- □ *FTIE* — Exclusive tie
- □ *FUNTIE* — Untie (close) files

Native DOS Files:

- □ *NAPPEND* — Append APL data to regular DOS file as byte stream
- □ *NCREATE* — Create DOS file
- □ *NERASE* — Erase DOS file
- □ *NNAMES* — List names of tied (open) DOS files
- □ *NNUMS* — List tie numbers of open DOS files
- □ *NREAD* — Read data from DOS file into workspace

- ☐ *NREPLACE* — Replace date in DOS file
- ☐ *NRESIZE* — Change size of DOS file
- ☐ *NSIZE* — Report size of DOS file
- ☐ *NTIE* — Open DOS file
- ☐ *NUNTIE* — Untie (close) DOS files

Saved Workspaces:

- ☐ *COPY* — Copy objects under program control
- ☐ *LOAD* — Load workspace under program control
- ☐ *PCOPY* — Protective copy under program control
- ☐ *QLOAD* — Load workspace (quietly) under program control
- ☐ *WSLIB* — List saved workspaces in a directory
- ☐ *XLOAD* — Load workspace without executing latent expression

Active Workspace Environment:

- ☐ *AI* — Accounting Information
- ☐ *CR* — Canonical representation of a function
- ☐ *CRL* — Canonical representation of one line of a function
- ☐ *CRLPC* — Extract public comment from a function
- ☐ *CT* — Comparison tolerance
- ☐ *DEF* — Define a function from character form
- ☐ *DEFL* — Replace/insert line in a function
- ☐ *DL* — Delay for specific interval
- ☐ *DR* — Report or change internal data type for variable
- ☐ *ERASE* — Erase a function or variable
- ☐ *EX* — Erase a function or variable
- ☐ *FX* — Define a function from canonical form
- ☐ *IDLIST* — List objects in a workspace
- ☐ *IDLOC* — Report definitions of localized identifiers at all stack levels
- ☐ *IO* — Index origin
- ☐ *LC* — Line counter
- ☐ *LOCK* — Lock a function under program control
- ☐ *LX* — Latent expression
- ☐ *NC* — Name class of an APL object
- ☐ *NL* — List objects of a given class in workspace
- ☐ *PP* — Print precision
- ☐ *PW* — Print width
- ☐ *RL* — Random number seed
- ☐ *SI* — Report current state indicator stack
- ☐ *SINL* — Display state indicator and localized names
- ☐ *SIZE* — Report size of an APL object
- ☐ *STOP* — Report or set stop vector on a function
- ☐ *SYMB* — Report current symbol table size and number used
- ☐ *TRACE* — Trace function execution
- ☐ *VR* — Visual representation of function
- ☐ *WA* — Amount of unused workspace area available
- ☐ *WSID* — Name of active workspace
- ☐ *WSSIZE* — Size of current active workspace
- ☐ *WSOWNER* — Identify owner of active workspace
- ☐ *WSTS* — Timestamp of active workspace

Exception Handling:

- ☐ *ALX* — Latent-expression handler for weak interrupts
- ☐ *DM* — Most recently produced error message
- ☐ *ELX* — Latent-expression handler for APL errors
- ☐ *ERROR* — Signal user-defined error under program control
- ☐ *SA* — Stop action: exits function, clears stack or exits APL upon immediate execution input

Formatting:

- ☐ *FI* — Input conversion of character to numeric
- ☐ *FMT* — General purpose number and text formatter
- ☐ *VI* — Input conversion verifier

Communications and Peripherals:

- ☐ *ARBIN* — General-purpose input/output function for communicating with peripherals and remote computers. Handles port selection, translation, overstriking, time-outs, buffer sizes, and terminator characters
- ☐ *ARBOUT* — Transmits arbitrary character sequences to screen
- ☐ *CAP* — Specifies input capture file

□ *CRT* — Information on screen size and scrolling memory size
□ *CURSOR* — Move cursor to position on screen or reports current position
□ *IN* — Specify input file (as keyboard replacement)
□ *INBUF* — Add or remove keystrokes from keyboard buffer
□ *INKEY* — Read a single keystroke from the keyboard buffer
□ *IOSTATE* — Information on input/output source
□ *OUT* — Specify output file
□ *PFKEY* — Read or set function keys
□ *WGET* — Get data from screen
□ *WIN* — Multi-window screen input processor
□ *WINDOW* — Read or set current active window
□ *WKEY* — Define keystroke behavior for □*WIN*
□ *WPUT* — Write data to screen

This just came in in Version --

Interface to DOS and non-APL Programs:
□ *CALL* — Call assembler routine in APL workspace
□ *CMD* — Execute DOS commands under program control
□ *INT* — Call a system interrupt
□ *LIB* — Report all files in a directory, regardless of type
□ *LIBS* — Report current directory definitions of numbered APL libraries
□ *PEEK* — Retrieve value of memory location
□ *POKE* — Replace memory location with new value
□ *SEG* — Set, or retrieve, segment register value
□ *STPTR* — Locate memory address of an APL object

New features came in in Version 6

Graphics Functions:
□ *GCIRCLE* — Draw circle, arcs, or ellipses
□ *GINIT* — Initialize graphics boards
□ *GLINE* — Draw lines, points, and rectangles
□ *GMASK* — Specify mask pixels when drawing lines or circles
□ *GPAINT* — Fill in region of screen in specified color
□ *GPRINT* — Copy graphics image to printer
□ *GSHADE* — Shade rectangles with color or patterns
□ *GSTATUS* — Provide information on graphics display parameters
□ *GTYPE* — Select or report the type of pixel interaction when drawing new lines
□ *GVIEW* — Select or report the area of the screen to project graphics image
□ *GWINDOW* — Select or report the portion of the graphics easel to display on the screen
□ *GWRITE* — Write characters on graphics screen
□ *GZOOM* — Set magnification factor for graphics characters

Miscellaneous:
□ *AV* — Atomic vector
□ *EDIT* — Full-screen function and data editor
□ *HELP* — Call HELP facility
□ *HNAMES* — Specify HELP files to be used
□ *HTOPICS* — Specify topic to be searched for when the HELP facility is activated
□ *KEYW* — Set or retrieve keyword state
□ *SOUND* — Play music on system speaker
□ *SS* — String search function
□ *SYSID* — APL★PLUS System identification
□ *TC* — Terminal control constants (also □*TCBEL*, □*TCBS*, □*TCDEL*, □*TCESC*, □*TCFF*, □*TCLF*, □*TCNL*, □*TCNUL*)
□ *TS* — Current time stamp
□ *UL* — Report number of users (not activated)

System Commands:
) — Recall last APL statement for editing
)CLEAR — Clear workspace
)CMD — Execute DOS command
)COPY — Copy from saved workspace
)DROP — Delete saved workspace
)EDIT — Edit function or data
)ERASE — Erase APL objects
)FLIB — List component files in a directory
)FNS — List functions in workspace
)HELP — Activate HELP facility
)LIB — List all files in a directory
)LIBS — List all current directories available
)LOAD — Load a saved workspace
)OFF — End APL session
)PCOPY — Copy from saved workspace without replacement
)RESET — Clear state indicator
)SAVE — Save the active workspace
)SI — Display state indicator
)SIC — Clear state indicator

)SINL	— Display state indicator and localized names
)STATLINE	— Turn status line on or off
)SYMBOLS	— Display or change size of symbol table
)VARS	— List variables in a workspace
)WSID	— Display or change name of active workspace
)WSLIB	— List workspaces in a directory
)XLOAD	— Load workspace suppressing execution of latent expression

SYSTEM REQUIREMENTS

The APL★PLUS PC System is designed for the IBM Personal Computer or compatibles equipped with the PC DOS or MS-DOS® operating system, at least 256K bytes of RAM memory, and at least one double-sided disk drive.

APL characters are supported on both the monochrome, the color, and the enhanced color display adapters. Other popular display adapters are also supported.

Any printer compatible with the personal computer will work with the APL★PLUS PC system. Only some printers, however, can print the APL symbols or print screen graphics. Epson® compatible graphics printers will print both APL graphics and APL symbols. Other printers are supported for printing APL symbols through down-load character sets. Printers with built-in APL character sets are supported directly.

PRODUCT CONTENTS

The APL★PLUS PC System consists of:

- Complete documentation in custom linen binders and boxes
- Character ROM or software to produce APL characters
- APL keyboard labels and keyboard placard
- Three diskettes containing system software and utilities
- An APL tutorial and popular textbook

PORTABILITY

The APL★PLUS PC System is functionally compatible with STSC's other APL★PLUS Systems running in the TSO, CMS, and the UNIX™ environments. Built-in communications software allows the PC to transfer data and programs to and from computers.

AVAILABILITY

STSC's APL★PLUS Systems are available through local dealers or by contacting STSC, Inc. An APL★PLUS PC Run-Time System is available for application developers.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

APL★PLUS is a registered trademark of STSC, Inc. PLUS★WARE is a trademark of STSC, Inc. Epson, IBM, and MS-DOS are registered trademarks of Epson Corporation, International Business Machines, and Microsoft, respectively. UNIX is a trademark of AT&T Bell Laboratories.

Copyright © 1985, STSC, Inc.
Printed in the U.S.A.
CC188–1185

A PLUS★WARE™ PRODUCT

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, MD 20852
301-984-5000

Software Publishing Group

Dear APL*PLUS PC System Update Customer,

Congratulations! You have just received your Release 6 update to the APL*PLUS System for the PC. The enclosed brochure highlights some of the new features in this release. Most of these features reflect suggestions you or other customers have made. Thank you for the feedback. Your comments help us to continue to improve the APL*PLUS PC System.

I'm sure you're anxious to try out your new system's improvements and features. But before you do, make sure you received everything. In your package, you should find:

- Completely new, reorganized documentation. Replace all your current documentation, except standalone texts like APL Is Easy!, with the new documentation. The new documentation is much more concise than the old, so you should find plenty of room in your binders for it. The reorganization consolidated the information in the previous manuals to make it easier to find and understand; for example, error messages were collected in one section rather than repeated throughout the manual. Other topics that were covered in several places throughout the documentation have been consolidated into tutorial books.

  If you have made notes or annotated your old documentation, you may wish to retain these sections until you're familiar with the new organization.

- A Getting Started book and a Quick Reference Guide. Both of these books are new for this release.

- Three diskettes. These diskettes contain the complete Release 6 system. However, some of the workspaces supplied with the system have not changed since Release 5.0 and are not included with the update.

The most significant change to the APL*PLUS PC System in Release 6 is the relaxation of the 64K size limit and 32K dimension limit. The maximum variable size was increased by creating three new variable types: large integer, floating point, and character variables. These changes mean that all your old Assembler language functions written to work with limited variables should still work without modification. However, we have changed the syntax of ⎕*STPTR* to prevent old assembler programs from being able to access new, larger variables unless

(over)

the programs are updated to reflect the new syntax. This will keep old assembler programs from damaging the workspace if you attempt to use them with large variables.

1 ⎕*POKE* 303 has been added to Release 6 to prevent the system from creating the new, larger objects. If you plan to move data from Release 6 back to an earlier release, you should use this expression. If you don't, you could create objects larger than those acceptable in the earlier release. The results are disastrous:

-)*LOAD* and)*COPY* from an earlier release will not work correctly on Release 6 workspaces that contain large variables.
- Files with large components will either give *FILE DAMAGED* reports or ⎕*FREAD* will produce a damaged variable if used with earlier releases.

In a few instances, primitive functions were not extended to handle larger objects if the result would be so large as to be unusable in a 512K workspace size. For example, ⍋, and ⍒ cannot produce meaningful results where the result would have a dimension greater than 32K. Likewise, ⌹ retains the previous 64K restrictions on its arguments.

Another difference to note is that the default working directories are specified differently in the APLLIBS file and in the ⎕*LIBS* list than in previous releases.

STSC is committed to providing quality software packages. If you have problems using this package that you can't resolve through our Help Line, or if you have suggestions for improving the system, please write to me. We want you to be happy with your system and with our service. (Of course, if you love the system, I'd enjoy hearing from you, too.)

We are proud of the enhancements we have provided to microcomputer APL users. We hope you enjoy using Release 6 as much as we enjoyed making it for you.

Sincerely,

Ed Myers

Edward R. Myers
APL Products Marketing Manager

reference guide provides an exhaustive list of the syntax and parameters for the language features of this system.

RELEASE 6.0 CONTENTS

The Release 6.0 update package includes:

- One serialized system disk and two utility disks,
- Complete system documentation, and the
- New Pocket Reference Guide.

SYSTEM REQUIREMENTS

The APL★PLUS PC System, Release 6.0 runs on IBM PC or compatible microcomputers with a monochrome, color, or EGA display adapter. One double-sided disk drive, 256K RAM memory, and DOS 2.0 or later are required.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852

301-984-5123

APL★PLUS is a registered trademark and service mark of STSC, Inc. IBM is a registered trademark of International Business Machines. PLUS★WARE is a trademark of STSC, Inc. and Hewlett Packard is a trademark of Hewlett Packard Corporation.

CCP99-0786
Printed in the U.S.A.

A PLUS★WARE™ PRODUCT

BREAKING THE 64K BARRIER!

Certainly the most exciting feature of Release 6.0 of the APL★PLUS PC System is unrestricted variable size. In previous releases, variables were limited to a maximum size of 64K and, in some cases, 32K. With more and more users encountering these limits, STSC invested substantial time and effort to break this barrier.

In Release 6.0, the size of a variable is limited only by the available room in your workspace. The addition of large variables does not affect performance when using the system with variables less than 64K in size.

LASER PRINTER SUPPORT
Support is now available for the Hewlett-Packard™ LaserJet+ printer. With Release 6.0 and the HP laser printer, you can print a mixture of APL and ASCII characters in near-typeset quality. In addition, graphics routines have been modified to allow extremely high resolution (1024 X 1024 pixels) output to the laser printer. Now, resolution quality on the HP is superior to the 640 X 350 resolution available on the highest resolution display adapters supported with Release 5.0 of the APL*PLUS PC System.

GRAPHICAL CHARACTER SET ENHANCEMENTS
With previous releases, the APL characters could be generated graphically on IBM® color card-compatible graphics systems. This allowed use of the APL*PLUS PC System on machines without replacement of a character ROM. These graphical characters, however, prevented the use of two important productivity features: screen scrolling and the full-screen editor.

With Release 6.0 enhancements, the graphical characters are displayed over 6 times faster. With this additional display speed it was possible to modify the full-screen editor and screen scrolling to work in graphics mode. The graphical character routines also support the Hercules and compatible cards in addition to the IBM color cards. This means that APL is now fully supported on all 100% compatible computers—even without an APL character ROM.

IMPROVED FILE SYSTEM
Users may now add and delete DOS directories from the table of recognized APL libraries without leaving the APL session by using the new Release 6.0 features *⎕LIBD*, *⎕CHDIR*, *⎕MKDIR*, and *⎕RMDIR*.

Native files can now be specified with the complete DOS path name in addition to the short-hand pseudo-drive designation used in previous releases.

ENHANCEMENTS TO ⎕WIN
First released with Release 5.0 of the APL*PLUS PC System, *⎕WIN* has proven to be a powerful and well-used feature. *⎕WIN* allows system developers to provide fast menus and full-screen forms-style entry to APL applications.

With Release 6.0 enhancements, *⎕WIN* includes enhancements suggested by APL*PLUS PC System users, such as new key actions and better field validation. For example, the PrtSc, CapsLock, and NumLock keys are now recognized by *⎕WIN*. Numeric validation has been enhanced to accept negative numbers using the minus sign (−) in addition to the APL "high bar" (¯) symbol. Special key actions allow for fast processing of multi-screen formats.

COMPLEX NUMBERS AND EIGENVALUE FUNCTIONS
Added with Release 6.0 are utility workspaces containing special functions to perform complex number calculations, determine roots of complex functions, and compute eigenvalues. These functions will assist scientists, engineers, and others who need these advanced mathematical features.

NEW DOCUMENTATION AND POCKET REFERENCE GUIDE
System documentation has been modified, reorganized, and new sections added to provide an easier-to-use format. And, there is no need to update your existing manuals as all new documentation is provided with the update.

A must for the mobile APL*PLUS System user, a convenient pocket reference guide is now included with the APL*PLUS PC System. The

APL★PLUS® PC System

VERSION 6

STSC's APL★PLUS PC System is the most advanced application development system available for the microcomputer. It gives problem-solvers the power to deliver totally integrated, highly efficient solutions to problems that conventional software simply can't handle.

Because of its efficiency, you can develop programs four to ten times faster than with conventional languages. The APL★PLUS PC System can be integrated with other applications you already have, or used to link those that aren't "speaking" to each other.

SYSTEM FEATURES

- A complete APL★PLUS implementation in an extended working environment
- Traditional APL character set or English-like keywords
- Workspaces up to 500K bytes
- Math Coprocessor Support (if installed)
- On-line HELP facility
- High resolution graphics supporting many popular graphics cards, including:
  - — IBM Enhanced Graphics Adapter
  - — IBM Color Adapter
  - — Hercules Graphics Adapters
  - — TECMAR Graphics Master
- Built-in terminal emulation and Upload/Download communications capability
- Full-screen word processing style editor for functions or character data
- Spreadsheet-like editor for numeric data
- Multi-window input manager
- Off-screen scrolling
- User definable function keys
- Programmable keyboard buffer
- Status line
- Interfaces to non-APL programs and packages
- Ability to execute DOS commands directly
- Component file system and interfaces to native DOS files
- Report-formatting facility
- Error-trapping features
- Over 100 system functions and over 200 APL and assembler utility functions
- Extensive documentation.

SYSTEM SPECIFICATIONS AND LIMITS

DATA TYPES

Integer: 16 bits per element
Floating Point: 64-bit IEEE Standard double precision
Character: 8 bits per element

SYSTEM LIMITS

Numeric range: −1.797E308 to 1.797E308
Maximum rank: 63
Maximum variable size: workspace size
Maximum coordinate length: unlimited
Maximum lines in function: 9999
Maximum symbols: 2048

SYSTEM FUNCTIONS

APL Component Files:

- ☐ *FAPPEND* — Append array to component file
- ☐ *FAVAIL* — Report file system availability (not activated)
- ☐ *FCREATE* — Create component file
- ☐ *FDROP* — Drop components from file
- ☐ *FDUP* — Duplicate file and compact unused space
- ☐ *FERASE* — Erase component file
- ☐ *FHOLD* — Hold File (not activated)
- ☐ *FLIB* — List component files in a directory
- ☐ *FNAMES* — List names of tied files
- ☐ *FNUMS* — List tie numbers of tied files
- ☐ *FRDAC* — Read file access matrix
- ☐ *FRDCI* — Read audit trail information on file component
- ☐ *FREAD* — Read component from file into workspace
- ☐ *FRENAME* — Rename component file
- ☐ *FREPLACE* — Replace component
- ☐ *FRESIZE* — Change size limit on file
- ☐ *FSIZE* — Report file size information
- ☐ *FSTAC* — Set file access matrix
- ☐ *FSTIE* — Share tie (same as ☐ *FTIE*)
- ☐ *FTIE* — Exclusive tie
- ☐ *FUNTIE* — Untie (close) files

Native DOS Files:

- ☐ *NAPPEND* — Append APL data to regular DOS file as byte stream
- ☐ *NCREATE* — Create DOS file
- ☐ *NERASE* — Erase DOS file
- ☐ *NNAMES* — List names of tied (open) DOS files
- ☐ *NNUMS* — List tie numbers of open DOS files
- ☐ *NREAD* — Read data from DOS file into workspace
- ☐ *NRENAME* — Rename a DOS file

- ☐ *NREPLACE* — Replace date in DOS file
- ☐ *NRESIZE* — Change size of DOS file
- ☐ *NSIZE* — Report size of DOS file
- ☐ *NTIE* — Open DOS file
- ☐ *NUNTIE* — Untie (close) DOS files

Saved Workspaces:

- ☐ *COPY* — Copy objects under program control
- ☐ *LOAD* — Load workspace under program control
- ☐ *PCOPY* — Protective copy under program control
- ☐ *QLOAD* — Load workspace (quietly) under program control
- ☐ *WSLIB* — List saved workspaces in a directory
- ☐ *XLOAD* — Load workspace without executing latent expression

Active Workspace Environment:

- ☐ *AI* — Accounting Information
- ☐ *CR* — Canonical representation of a function
- ☐ *CRL* — Canonical representation of one line of a function
- ☐ *CRLPC* — Extract public comment from a function
- ☐ *CT* — Comparison tolerance
- ☐ *DEF* — Define a function from character form
- ☐ *DEFL* — Replace/insert line in a function
- ☐ *DL* — Delay for specific interval
- ☐ *DR* — Report or change internal data type for variable
- ☐ *ERASE* — Erase a function or variable
- ☐ *EX* — Erase a function or variable
- ☐ *FX* — Define a function from canonical form
- ☐ *IDLIST* — List objects in a workspace
- ☐ *IDLOC* — Report definitions of localized identifiers at all stack levels
- ☐ *IO* — Index origin
- ☐ *LC* — Line counter
- ☐ *LOCK* — Lock a function under program control
- ☐ *LX* — Latent expression
- ☐ *NC* — Name class of an APL object
- ☐ *NL* — List objects of a given class in workspace
- ☐ *PP* — Print precision
- ☐ *PW* — Print width
- ☐ *RL* — Random number seed
- ☐ *SI* — Report current state indicator stack
- ☐ *SINL* — Display state indicator and localized names
- ☐ *SIZE* — Report size of an APL object
- ☐ *STOP* — Report or set stop vector on a function
- ☐ *SYMB* — Report current symbol table size and number used
- ☐ *TRACE* — Trace function execution
- ☐ *VR* — Visual representation of function
- ☐ *WA* — Amount of unused workspace area available
- ☐ *WSID* — Name of active workspace
- ☐ *WSSIZE* — Size of current active workspace
- ☐ *WSOWNER* — Identify owner of active workspace
- ☐ *WSTS* — Timestamp of active workspace

Exception Handling:

- ☐ *ALX* — Latent-expression handler for weak interrupts
- ☐ *DM* — Most recently produced error message
- ☐ *ELX* — Latent-expression handler for APL errors
- ☐ *ERROR* — Signal user-defined error under program control
- ☐ *SA* — Stop action: exits function, clears stack or exits APL upon immediate execution input

Formatting:

- ☐ *FI* — Input conversion of character to numeric
- ☐ *FMT* — General purpose number and text formatter
- ☐ *VI* — Input conversion verifier

Communications and Peripherals:

- ☐ *ARBIN* — General-purpose input/output function for communicating with peripherals and remote computers. Handles port selection, translation, overstriking, time-outs, buffer sizes, and terminator characters
- ☐ *ARBOUT* — Transmits arbitrary character sequences to screen
- ☐ *CAP* — Specifies input capture file

□ *CRT* — Information on screen size and scrolling memory size
□ *CURSOR* — Move cursor to position on screen or reports current position
□ *IN* — Specify input file (as keyboard replacement)
□ *INBUF* — Add or remove keystrokes from keyboard buffer
□ *INKEY* — Read a single keystroke from the keyboard buffer
□ *IOSTATE* — Information on input/output source
□ *OUT* — Specify output file
□ *PFKEY* — Read or set function keys
□ *WGET* — Get data from screen
□ *WIN* — Multi-window screen input processor
□ *WINDOW* — Read or set current active window
□ *WKEY* — Define keystroke behavior for □*WIN*
□ *WPUT* — Write data to screen

Interface to DOS and non-APL Programs:
□ *CALL* — Call assembler routine in APL workspace
□ *CHDIR* — Change current working directory
□ *CMD* — Execute DOS commands under program control
□ *INT* — Call a system interrupt
□ *LIB* — Report all files in a directory, regardless of type
□ *LIBD* — Assign a library number to a DOS directory
□ *LIBS* — Report current directory definitions of numbered APL libraries
□ *MKDIR* — Make a new DOS directory
□ *PEEK* — Retrieve value of memory location
□ *POKE* — Replace memory location with new value
□ *RMDIR* — Remove a DOS directory
□ *SEG* — Set, or retrieve, segment register value
□ *STPTR* — Locate memory address of an APL object

Graphics Functions:
□ *GCIRCLE* — Draw circle, arcs, or ellipses
□ *GINIT* — Initialize graphics boards
□ *GLINE* — Draw lines, points, and rectangles
□ *GMASK* — Specify mask pixels when drawing lines or circles
□ *GPAINT* — Fill in region of screen in specified color
□ *GPRINT* — Copy graphics image to printer
□ *GSHADE* — Shade rectangles with color or patterns
□ *GSTATUS* — Provide information on graphics display parameters
□ *GTYPE* — Select or report the type of pixel interaction when drawing new lines
□ *GVIEW* — Select or report the area of the screen to project graphics image
□ *GWINDOW* — Select or report the portion of the graphics easel to display on the screen
□ *GWRITE* — Write characters on graphics screen
□ *GZOOM* — Set magnification factor for graphics characters

Miscellaneous:
□ *AV* — Atomic vector
□ *EDIT* — Full-screen function and data editor
□ *HELP* — Call HELP facility
□ *HNAMES* — Specify HELP files to be used
□ *HTOPICS* — Specify topic to be searched for when the HELP facility is activated
□ *KEYW* — Set or retrieve keyword state
□ *SOUND* — Play music on system speaker
□ *SS* — String search function
□ *SYSID* — APL★PLUS System identification
□ *TC* — Terminal control constants (also □*TCBEL*, □*TCBS*, □*TCDEL*, □*TCESC*, □*TCFF*, □*TCLF*, □*TCNL*, □*TCNUL*)
□ *TS* — Current time stamp
□ *UL* — Report number of users (not activated)

System Commands:
) — Recall last APL statement for editing
)CLEAR — Clear workspace
)CMD — Execute DOS command
)COPY — Copy from saved workspace
)DROP — Delete saved workspace
)EDIT — Edit function or data
)ERASE — Erase APL objects
)FLIB — List component files in a directory
)FNS — List functions in workspace
)HELP — Activate HELP facility
)LIB — List all files in a directory
)LIBS — List all current directories available
)LOAD — Load a saved workspace
)OFF — End APL session
)PCOPY — Copy from saved workspace without replacement

)RESET — Clear state indicator
)SAVE — Save the active workspace
)SI — Display state indicator
)SIC — Clear state indicator
)SINL — Display state indicator and localized names
)STATLINE — Turn status line on or off
)SYMBOLS — Display or change size of symbol table
)VARS — List variables in a workspace
)WSID — Display or change name of active workspace
)WSLIB — List workspaces in a directory
)XLOAD — Load workspace suppressing execution of latent expression

SYSTEM REQUIREMENTS

The APL★PLUS PC System is designed for the IBM Personal Computer or compatibles equipped with the PC DOS or MS-DOS® operating system, at least 256K bytes of RAM memory, and at least one double-sided disk drive.

APL characters are supported on both the monochrome, the color, and the enhanced color display adapters. Other popular display adapters are also supported.

Any printer compatible with the personal computer will work with the APL★PLUS PC system. Only some printers, however, can print the APL symbols or print screen graphics. Epson® compatible graphics printers will print both APL graphics and APL symbols. Other printers including the HP LaserJet+ are supported for printing APL symbols through down-load character sets. Printers with built-in APL character sets are supported directly.

PRODUCT CONTENTS

The APL★PLUS PC System consists of:

- Complete documentation in custom linen binders and boxes
- Character ROM or software to produce APL characters
- APL keyboard labels and keyboard placard
- Four diskettes containing system software and utilities
- An APL tutorial and popular textbook

PORTABILITY

The APL★PLUS PC System is functionally compatible with STSC's other APL★PLUS Systems running in the TSO, CMS, and the UNIX™ environments. Built-in communications software allows the PC to transfer data and programs to and from computers.

AVAILABILITY

STSC's APL★PLUS Systems are available through local dealers or by contacting STSC, Inc. An APL★PLUS PC Run-Time System is available for application developers.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)

APL★PLUS is a registered trademark of STSC, Inc. PLUS★WARE is a trademark of STSC, Inc. Epson, IBM, and MS-DOS are registered trademarks of Epson Corporation, International Business Machines, and Microsoft, respectively. UNIX is a trademark of AT&T Bell Laboratories.

Copyright © 1986, STSC, Inc.
Printed in the U.S.A.
CC188-0886

A PLUS★WARE™ PRODUCT

APL★PLUS® PC System Version 7

STSC's APL★PLUS PC System is the most popular APL application development system available for the industry standard PC. It gives problem-solvers the power to deliver totally integrated, highly efficient solutions to problems that conventional software simply can't handle.

Because of its efficiency, you can develop programs four to ten times faster than with conventional languages. The APL★PLUS PC System can be integrated with other applications you already have, or used to link those that aren't "speaking" to each other.

SYSTEM FEATURES

- A complete APL★PLUS implementation in an extended working environment
- Traditional APL character set or English-like keywords
- Main workspaces up to 500K bytes
- Virtual Workspace Facility automatically swaps objects to and from file of up to 32 megabytes
- Math coprocessor support (if installed)
- On-line HELP facility
- High resolution graphics supporting many popular graphics cards, including:
  - — IBM PS/2 VGA
  - — IBM Enhanced Graphics Adapter
  - — IBM Color Adapter
  - — Hercules Graphics Adapters
  - — TECMAR Graphics Master
- Built-in terminal emulation and Upload/Download communications capability
- Full-screen word processing style editor for functions or character data
- Spreadsheet-like editor for numeric data
- Multi-window input manager
- Off-screen scrolling
- User definable function keys
- Programmable keyboard buffer
- Status line
- Interfaces to non-APL programs and packages
- Ability to execute DOS commands directly
- Component file system and interfaces to native DOS files
- Local area network support for concurrent sharing of files and workspaces
- Report-formatting facility
- Error-trapping features
- Over 100 system functions and over 200 APL and Assembler utility functions
- Traditional APL or ASCII-like unified keyboard layout
- Extensive documentation.

SYSTEM SPECIFICATIONS AND LIMITS

DATA TYPES

Integer: 16 bits per element
Floating Point: 64-bit IEEE Standard double precision
Character: 8 bits per element

SYSTEM LIMITS

Numeric range: -1.797E308 to 1.797E308
Maximum rank: 63
Maximum variable size: size of main workspace
Maximum coordinate length: unlimited
Maximum lines in function: 9999
Maximum symbols: 2048

SYSTEM FUNCTIONS

APL Component Files:

⎕FAPPEND	— Append array to component file
⎕FAVAIL	— Report file system availability
⎕FCREATE	— Create component file
⎕FDROP	— Drop components from file
⎕FDUP	— Duplicate file and compact unused space
⎕FERASE	— Erase component file
⎕FHOLD	— Hold file across network
⎕FLIB	— List component files in a directory
⎕FNAMES	— List names of tied (open) files
⎕FNUMS	— List tie numbers of tied files
⎕FRDAC	— Read file access matrix
⎕FRDCI	— Read audit trail information on file component
⎕FREAD	— Read component from file into workspace
⎕FRENAME	— Rename component file
⎕FREPLACE	— Replace component
⎕FRESIZE	— Change size limit on file
⎕FSIZE	— Report file size information
⎕FSTAC	— Set file access matrix
⎕FSTIE	— Share tie across network
⎕FTIE	— Exclusive tie across network
⎕FUNTIE	— Untie (close) files

Native DOS Files:

⎕NAPPEND	— Append APL data to regular DOS file as byte stream
⎕NCREATE	— Create DOS file
⎕NERASE	— Erase DOS file
⎕NNAMES	— List names of tied (open) DOS files
⎕NNUMS	— List tie numbers of open DOS files
⎕NREAD	— Read data from DOS file into workspace
⎕NRENAME	— Rename a DOS file
⎕NREPLACE	— Replace data in DOS file
⎕NRESIZE	— Change size of DOS file
⎕NSIZE	— Report size of DOS file
⎕NTIE	— Open DOS file
⎕NUNTIE	— Untie (close) DOS files

Saved Workspaces:

⎕COPY	— Copy objects under program control
⎕LOAD	— Load workspace under program control
⎕PCOPY	— Protective copy under program control
⎕QLOAD	— Load workspace (quietly) under program control
⎕WSLIB	— List saved workspaces in a directory
⎕XLOAD	— Load workspace without executing latent expression

Active Workspace Environment:

⎕AI	— Accounting information
⎕CR	— Canonical representation of a function
⎕CRL	— Canonical representation of one line of a function
⎕CRLPC	— Extract public comment from a function
⎕CT	— Comparison tolerance
⎕DEF	— Define a function from character form
⎕DEFL	— Replace/insert line in a function
⎕DL	— Delay for a specific interval
⎕DR	— Report or change internal data type for variable
⎕ERASE	— Erase a function or variable
⎕EX	— Erase a function or variable
⎕FX	— Define a function from canonical form
⎕IDLIST	— List objects in a workspace
⎕IDLOC	— Report definitions of localized identifiers at all stack levels
⎕IO	— Index origin
⎕LC	— Line counter
⎕LOCK	— Lock a function
⎕LX	— Latent expression
⎕MF	— Monitor function executions
⎕NC	— Name class of an APL object
⎕NL	— List objects of a given class in workspace
⎕PP	— Print precision
⎕PW	— Print width
⎕RL	— Random number seed
⎕SI	— Report current state indicator stack
⎕SINL	— Display state indicator and localized names
⎕SIZE	— Report size of an APL object
⎕STOP	— Report or set stop vector on a function
⎕SYMB	— Report current symbol table size and number used
⎕TRACE	— Trace function execution
⎕VR	— Visual representation of function
⎕WA	— Amount of unused workspace area available
⎕WSID	— Name of active workspace
⎕WSSIZE	— Size of current active workspace
⎕WSOWNER	— Identify owner of active workspace
⎕WSTS	— Timestamp of active workspace

Exception Handling:

⎕ALX	— Latent-expression handler for weak interrupts
⎕DM	— Most recently produced error message
⎕ELX	— Latent-expression handler for APL errors
⎕ERROR	— Signal user-defined error under program control
⎕SA	— Stop action: exits function, clears stack, exits APL, or signals error upon return to immediate execution

Formatting:

⎕FI	— Conversion of character input to numeric
⎕FMT	— General purpose number and text formatter
⎕VI	— Input conversion verifier

Communications and Peripherals:

⎕ARBIN	— General-purpose input/output function for communicating with peripherals and remote computers. Handles port selection, translation, overstriking, time-outs, buffer sizes, and terminator characters
⎕ARBOUT	— Transmits arbitrary character sequences to screen
⎕CAP	— Specifies input capture file

⎕CRT	— Information on screen size and scrolling memory size
⎕CURSOR	— Move cursor to position on screen or reports current position
⎕IN	— Specify input file (as keyboard replacement)
⎕INBUF	— Add or replace keystrokes to keyboard buffer
⎕INKEY	— Read a single keystroke from the keyboard buffer
⎕IOSTATE	— Information on input/output source
⎕OUT	— Specify output file
⎕PFKEY	— Read or set function keys
⎕WGET	— Get data from screen
⎕WIN	— Multi-window screen input processor
⎕WINDOW	— Read or set current active window
⎕WKEY	— Define keystroke behavior for *⎕WIN*
⎕WPUT	— Write data to screen

Interface to DOS and non-APL Programs:

⎕CALL	— Call Assembler routine in APL workspace
⎕CHDIR	— Change current working directory
⎕CMD	— Execute DOS commands under program control
⎕INT	— Call a system interrupt
⎕LIB	— Report all files in a directory regardless of type
⎕LIBD	— Assign a library number to a DOS directory
⎕LIBS	— Report current directory definitions of numbered APL libraries
⎕MKDIR	— Make a new DOS directory
⎕PEEK	— Retrieve value of memory location
⎕POKE	— Replace memory location with new value
⎕RMDIR	— Remove a DOS directory
⎕SEG	— Set or retrieve segment register value
⎕STPTR	— Locate memory address of an APL object

Graphics Functions:

⎕GCIRCLE	— Draw circle, arcs, or ellipses
⎕GINIT	— Initialize graphics boards
⎕GLINE	— Draw lines, points, and rectangles
⎕GMASK	— Specify mask pixels when drawing lines or circles
⎕GPAINT	— Fill in region of screen in specified color
⎕GPRINT	— Copy graphics image to printer
⎕GSHADE	— Shade rectangles with color or patterns
⎕GSTATUS	— Provide information on graphics display parameters
⎕GTYPE	— Select or report the type of pixel interaction when drawing new lines
⎕GVIEW	— Select or report the area of the screen to project graphics image
⎕GWINDOW	— Select or report the portion of the graphics easel to display on the screen
⎕GWRITE	— Write characters on graphics screen
⎕GZOOM	— Set magnification factor for graphics characters

Miscellaneous:

⎕AV	— Atomic vector
⎕EDIT	— Full-screen function and data editor
⎕HELP	— Call HELP facility
⎕HNAMES	— Specify HELP files to be used
⎕HTOPIC	— Specify topic to be searched for when the HELP facility is activated
⎕KEYW	— Set or retrieve keyword state
⎕SOUND	— Play music on system speaker
⎕SS	— String search function
⎕SYSID	— APL★PLUS System identification
⎕TC	— Terminal control constants (also *⎕TCBEL. ⎕TCBS. ⎕TCDEL. ⎕TCESC. ⎕TCFF. ⎕TCLF. ⎕TCNL. ⎕TCNUL. ⎕DTCHT*)
⎕TS	— Current time stamp
⎕UL	— Report number of users

SYSTEMS COMMANDS

)	— Recall last APL statement for editing
)CLEAR	— Clear workspace
)CMD	— Execute DOS command
)COPY	— Copy from saved workspace
)DROP	— Delete saved workspace
)EDIT	— Edit function or data
)ERASE	— Erase APL objects
)FLIB	— List component files in a directory
)FNS	— List functions in workspace
)HELP	— Activate HELP facility
)LIB	— List all files in a directory
)LIBS	— List all current directories available
)LOAD	— Load a saved workspace
)OFF	— End APL session
)PCOPY	— Copy from saved workspace without replacement

)RESET	— Clear state indicator
)SAVE	— Save the active workspace
)SI	— Display state indicator
)SIC	— Clear state indicator
)SINL	— Display state indicator and localized names
)STATLINE	— Turn status line on or off
)SYMBOLS	— Display or change size of symbol table
)VARS	— List variables in a workspace
)WSID	— Display or change name of active workspace
)WSLIB	— List workspaces in a directory
)XLOAD	— Load workspace suppressing execution of latent expression

VDI STANDARD GRAPHICS

- Interface to GSS*CGI graphics drivers
- VDI workspace makes all VDI functions available from APL
- Selected device drivers are included with product disks

SYSTEM REQUIREMENTS

The APL★PLUS PC System is designed for the IBM Personal Computer or compatibles equipped with the PC DOS or MS-DOS® operating system version 3.1 or later, at least 384 bytes of RAM memory, and at least one double-sided disk drive.

APL character fonts are provided for IBM EGA, PS/2 VGA, Hercules Graphics Card Plus, and others. APL Character generator ROMs are available for the IBM monochrome, and CGA card, and other popular display adapters.

Any printer compatible with the personal computer will work with the APL★PLUS PC system. Only some printers, however, can print the APL symbols or print screen graphics.

Epson® compatible graphics printers will print both APL graphics and APL symbols. Other printers including the HP LaserJet Plus are supported for printing APL symbols through down-loaded character sets. Printers with built-in APL character sets are supported directly.

PRODUCT CONTENTS

The APL★PLUS PC System consists of:

- Complete documentation
- Character ROM or software to produce APL characters
- APL keyboard labels and keyboard placard
- Seven diskettes containing system software and utilities
- An APL tutorial and popular textbook
- Additional GSS*CGI device drivers and the GSS*CGI programmer's manual are available at extra cost

PORTABILITY

The APL★PLUS PC System is generally compatible with STSC's other APL★PLUS Systems including those running on IBM mainframes under MVS/TSO and VM/CMS, VAX under VMS, and UNIX. Built-in communications software allows the PC to transfer data and programs to and from other computers.

AVAILABILITY

STSC's APL★PLUS Systems are available through local dealers or by contacting STSC, Inc. An APL★PLUS PC Run-Time System is available for application developers.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (In Maryland or Canada)
Telex #898085 (STSC ROVE)
Fax (301) 984-5094

APL★PLUS is a registered trademark of STSC, Inc. Epson, IBM, MS-DOS, Hercules, VAX and LaserJet Plus are registered trademarks of Epson Corporation, International Business Machines, Microsoft, Hercules Computer Technology and Digital Equipment Corporation respectively. UNIX is a trademark of AT&T Bell Laboratories. GSS*CGI is a trademark of Graphic Software Systems.

Copyright © 1988 STSC, Inc.
Printed in the U.S.A.
0688-2.5M

VERSION 8.0 NEW FEATURES SUMMARY

APL★PLUS® SYSTEM FOR THE PC

APL PROGRAMMING REACHES NEW HEIGHTS OF EFFICIENCY!

For years STSC's APL★PLUS PC System has set the pace for APL programming development. With the release of Version 8.0—The APL★PLUS Extended System—we now enter into a new realm of productivity.

Improve Your Productivity with the APL★PLUS PC Extended System

The centerpiece of Version 8.0 is the Extended System, which offers an additional APL environment consisting of the "Session Manager" and a special interpreter. Designed to be the friendliest and most productive "front end" ever for APL, the Session Manager combines the full-screen editor with the management of the APL session's input and output. The result is many features that enhance the interactive use of APL and make your programming more productive:

- Edit multiple objects at the same time. You can have a ring of editing sessions active at the same time and move between them and the APL session with ease.
- Retain many lines of your APL session. Depending on the amount of memory you allocate to the Session Manager, you can record hundreds of lines of screen output.
- True cut-and-paste editing of blocks of text or partial lines. You can move or copy text between editor sessions as well as the APL session. Copy APL output into a report you are editing. Test an APL statement, then move it into a function.
- Edit native DOS files with the APL full-screen editor. Record the APL session output in a native file with only a few keystrokes. Read a file into a character variable. The possibilities are endless.
- Control the Session Manager with pull-down menus or keystrokes. Menus make it easy to move between sessions and select editor options. Menus also work for terminal mode, letting you configure the terminal, set the baud rate, etc.

Dozens of New Ways to Make Your Job Easier!

With the release of Version 8.0, APL programming has never been easier. Dozens of new features included in the Session Manager allow you to increase your productivity—efficiently! You now have the capability to:

- Control function trace and stop from the editor.
- Use the token form of search and replace to find complete APL identifiers or numbers instead of substrings.
- Undo bulk deletions.
- Print a tagged block of text with a single keystroke.
- Use up to 999 PF keys that can be interactively defined.
- Preserve a session log with pull-down statement re-execution.
- Completely customize the keyboard.
- Use an expanded set of startup configuration parameters.
- Use screens larger than 25-by-80.
- Jump from an open parenthesis to the matching closing parenthesis with a single keystroke.

And you always have the option of operating under the APL★PLUS PC Standard System when your application requires a larger workspace area.

Organize Your Applications into Packages

The QPACK utility lets you pack a group of related functions and variables together into a single APL variable called a **package**. Packages can be stored in a component file or swapped to file with the virtual workspace facility, then brought back into the workspace and converted back to functions and variables very quickly.

Packages make it easy to keep sections of your application on file while they are not needed, so you can make the best use of available workspace.

Improved Virtual Workspace Facility

The Virtual Workspace Facilty introduced in Release 7.0 has been enhanced in Release 8.0. It is now possible for variables and functions that are smaller than 1024 bytes to be swapped to disk. You specify the minimum size of objects that are eligible for swapping.

Use APL★PLUS PC Under OS/2 and DOS 4.0

Release 8.0 has been modified to allow use in the DOS "compatibility box" under OS/2. DOS 4.0 is also fully supported.

More File Ties

You can tie up to 200 files at once using Release 8.0 with DOS 3.3 or higher.

Save Workspaces Under Program Control

The new system functions □SAVE and □PSAVE will save the active workspace without halting execution. With them, your applications can checkpoint their status at appropriate points for later recovery.

Code Page Support for Display Fonts

Release 8.0 lets you use the DOS code page facility to display APL characters on EGA and VGA, and use national language character sets with APL★PLUS PC. A utility workspace lets you edit your own character sets. With code page support, you can produce APL characters and other fonts with GSS*CGI (VDI) graphics.

Workspace Interchange Convention

The Source Level Transfer workspace implements the Workspace Interchange Convention described in the ANSI/ISO standard for APL, making it easier to move applications between the APL★PLUS PC and other APL systems.

And Much More . . .

- □INBUF can be used with the DOS keyboard.
- Overstrikes are supported by □WIN.
- □FSIZE estimates the amount of unused file space recoverable with □FDUP.
- □CALL and □INT optionally return the flags and segment registers.
- You can load suspended workspaces saved with different DOS memory configurations.
- The NEDIT workspace supports user-defined row and column labels.
- The EIGENVAL workspace generates complex-valued eigenvectors. The FFT workspace generates Fast Fourier Transforms.

Version 8.0 Contents

The Version 8.0 package includes:

- Updated documentation complete with a page replacement guide.
- A new **Getting Started** booklet, **Quick Reference Guide** and the **Extended System Handbook**.
- Nine diskettes to replace your present system diskettes.

Version 8.0 System Requirements

The APL★PLUS System for the PC, Version 8.0, runs on IBM PC, XT, AT, PS/2, or compatible microcomputers with a monochrome, color, EGA, or VGA display adapter. One double-sided disk drive, 384K RAM memory, and DOS 2.0 or later are required. Hard disk, 640K, and a math co-processor are recommended. For network support DOS 3.1 or later is required.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
800-592-0050
301-984-5123 (In Maryland)

Outside the U.S.A.:
301-984-5412
Telex 898085
Telefax 301-984-5094

APL★PLUS is a registered trademark of STSC, Inc.

Other product names mentioned in this brochure may also be trademark used here for identification only.

Copyright © 1988, STSC, Inc.
108815M
Printed in the U.S.A.

VERSION 9.0 ENHANCEMENTS

APL★PLUS® SYSTEM FOR THE PC

Newest APL Release Offers Improved Productivity, Ease of Use

Twenty years of STSC commitment to APL excellence stand behind the latest release of the APL★PLUS System for the PC. With the release of Version 9.0, APL programming on the PC has become even easier and more productive.

New memory management capabilities, numeric editing, a user command processor and all-new documentation are among the major enhancements that make Version 9.0 the best APL★PLUS PC software ever.

Memory Magic

With Version 9.0, you have two new ways to move the APL★PLUS system code outside your PC's normal 640K DOS region on 80286 and 80386 machines. As a result, a total of 128K of additional work space is available in the Standard System, and as much as 150K is available in the Extended System. This means that with the right combination of hardware and software, you can build even larger APL applications.

With previous versions of APL★PLUS PC, you could put 64K of code into expanded memory, using an EMS driver. Now, with a suitable Extended Memory Specification (XMS) driver and new sign-on parameters in Version 9.0, you can put an additional 64K of the APL interpreter into the high memory area and another 64K of the session manager into upper memory on your PC. (The third 64K memory block is only usable in the Extended System.) Software drivers suitable for use with the new sign-on parameters include 386MAX™ from Qualitas Inc., and HIMEM.SYS from Microsoft.

Full-Screen Editing of Numeric Arrays

Using the enhanced full-screen editor in Version 9.0, you can edit numeric matrices just as easily as you now edit functions or character arrays. Numeric editing is fully integrated into both the Standard and Extended Systems. You can search and replace, cut and paste, add or delete rows and columns, pull in arrays from the workspace, or select subarrays to save into variables.

User Command Processor

The new user command processor in Version 9.0 lets you use utilities like you use system commands. These utilities are always available without having them resident in the workspace. More than 25 helpful pre-defined commands are included in Version 9.0, allowing you quickly and easily to manipulate objects in the active workspace or in command files. For example, you can:

- Display function names, even those in saved workspaces
- Display the names of global and local variables in functions
- Locate the occurrences of a phrase in functions
- Compare the active workspace with a file
- Display documentation for commands

In addition, you can easily add your own commands to the system.

Other General System Enhancements

Other Version 9.0 enhancements designed to make your work easier and more productive are:

- Symbol Table Cleanup—Re-uses space taken by obsolete symbols, reducing the likelihood of a SYMBOL TABLE FULL message.
- Faster Loading and Copying of Workspaces—Improves system performance.
- Keystroke Help Screens in Separate Files—Gives you larger workspaces by reducing interpreter size.
- An Enhanced □FDUP Function—Preserves space in pre-sized files, allowing the recovery of unused space in the file on a component-by-component basis.
- New INT C8h (□INT 200) services for querying and purging the symbol table, reconciling the screen buffer, positioning the cursor, and more.
- New peeks and pokes for numeric editor tagging attributes and tab key direction, symbol table cleanup, switchable DOS keyboard, and more.

An Enhanced Standard System

When you use the APL★PLUS PC Standard System, Version 9.0 allows you to toggle between the APL and DOS keyboard drivers. This makes it easy to use or react to keystroke conventions of non-APL software while running APL.

A More Productive Extended System

The Extended System is even more productive with Version 9.0.

- A Point-and-Edit Feature makes it easy to trace program logic through subroutine calls when debugging. You just place the cursor on the name of the object that you want to edit. A single keystroke suspends the current editing session and begins editing the object. You don't even have to type the object's name!
- A Built-in Print Spooler lets you work while the session manager directs other output to the printer.
- The ability to create function key definitions in a configuration file.

New Printer and Graphics Support

The following printers are now supported:

- Epson LQ 850, LQ 1050, FX 850, and FX 1050
- Toshiba P321SL and P341SL
- HP DeskJet (requires 128K RAM cartridge) and HP DeskJet Plus

Version 9.0 also includes updated versions of GSS★CGI graphics drivers with enhanced configuration capabilities.

All-New Documentation

Version 9.0 includes more than 1,000 pages of completely new documentation—revised, reorganized, and reformatted to make APL★PLUS PC even easier to use. Included are:

- a new *Reference Manual*
- a new *User's Manual*
- an updated *Quick Reference Guide*
- new installation instructions

Other New Features

With Version 9.0, you also get:

- The HANDLERS Workspace, formerly available only as an add-on in PC TOOLS, contains an easy, consistent, self-documenting technique for incorporating reliable error-and-interrupt handling into your applications.
- New utilities for quick and easy line plots in the VDI Workspace. Scaling and labeling are done automatically.
- A memory and CPU diagnostic program that produces a report on the available memory in your PC.
- Improved installation procedures.

Version 9.0 System Requirements

The APL★PLUS System for the PC, Version 9.0, runs on IBM PC, XT, AT, PS/2, or compatible microcomputers with a monochrome, color, EGA, or VGA display adapter. One double-sided disk drive, 384K RAM memory and DOS 2.0 or later are required. Hard disk, 640K, and a math co-processor are recommended. For network support DOS 3.1 or later is required. For the "Memory Magic" features, an 80286 or 80386 machine with at least 64K of extended memory is required, along with an XMS driver. The system utilizes both EMS and XMS memory.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
(800) 592-0050
(301) 984-5123 (in Maryland)

Outside the U.S.A., contact your local APL★PLUS dealer, or call:
(301) 984-5412
Telex 898085
Telefax (301) 984-5094

APL★PLUS is a registered trademark of STSC, Inc.
Other product names mentioned in this brochure may also be trademarks, and are used here for identification only.

Copyright © 1989, STSC, Inc.
Printed in the U.S.A.
CC245-1089

VERSION 10 ENHANCEMENTS SUMMARY

APL★PLUS® SYSTEM FOR THE PC

Whether you are using APL★PLUS PC for ad-hoc problem solving or complete application development, you want to get results — fast! And, you want to enhance APL's inherent strengths using the latest advances in software and hardware technology to build applications that are more flexible and efficient and easier to use.

Version 10 of APL★PLUS PC was designed with your needs in mind. Take a look at the enhancements included in this newest release of the world's most popular micro APL system.

Interface To Non-APL Software For Increased Flexibility And Performance

Call subroutines written in C or FORTRAN with the new ⎕*NA* system function. Imagine how much capability you could add to your applications by interfacing APL★PLUS PC with other software like database managers, graphics packages, and scientific and statistical subroutine libraries. Or use ⎕*NA* to increase application performance with your own C or FORTRAN routines.

To get you started, Version 10 includes a demonstration workspace of sample routines (and their source code) that work with interfaces for Microsoft C (Version 5.1), Turbo C (Version 2.0), and Microsoft FORTRAN (Version 5.0).

Enhanced User Command Processor For Added Productivity

The User Command Processor, introduced in Version 9, lets you define and store a set of commands that are always available, regardless of the workspace you are using. With Version 10, the User Command Processor is built into the interpreter to give you immediate access to your user commands without having to use special APL functions or PF keys. You can access the User Command Processor with a "]" in immediate execution mode, or with the new ⎕*UCMD* system function under program control.

Version 10 also includes new user commands to report disk storage, compare user command files, and compare APL functions, variables, and DOS text files.

Support For The Latest Hardware And Software For Efficiency And Ease Of Use

- Windows 3.0. Version 10 includes improved APL font support under Microsoft® Windows™ 3.0. It also includes instructions for working with APL in more than one window, or running other software programs simultaneously.

- Mouse Support. You can now use a mouse in your development work, and build applications that can be controlled with a mouse. Take advantage of new point-and-click access to menus, move around the edit ring, scroll through sessions, and tag, copy, and move blocks of text or numbers just as you would with your keyboard. The IBM Personal System/2™ Mouse, Microsoft® Mouse, and other compatible devices are supported.

- Enhanced Color Monitor Support. Customize colors for your APL environment, including the status line, editing sessions, tagged regions, and the wrapped-line marker. Substitute high-intensity background colors for the blinking attribute on EGA and VGA adapters. A workspace of utility functions for effective use of colors is also included.

- Enhanced Text Mode Support. The 43- and 50-line text modes on EGA and VGA adapters are now supported, so you can view more lines and columns on a single screen.

- Enhanced APL Font Support. The APL font facility for EGA and VGA adapters has been enhanced to manage lower DOS memory more efficiently. It now also supports the

43- and 50-line text modes. And, you can now print APL characters on your PostScript® printer, and customize your APL output with a variety of PostScript options.

- New Graphics Device Support. Version 10 includes new GSS device drivers for the NEC P5 printer, PostScript printers, the latest IBM Proprinters, and high-resolution EGA and VGA display adapters. Most popular GSS drivers are also included with the system, and others are available in an optional Completer Kit. New GSS Version 2.16 software also includes a startup option to reserve sufficient memory for the drivers.

Additional Enhancements For Increased Productivity And Performance

- ⎕*FDUP* has been enhanced to provide significant performance improvements on files with large components.
- File system functions have been enhanced to let you reserve disk space for a component in advance. This feature helps eliminate wasted file space caused when you replace a small component with a larger one.
- ⎕*WIN* has been enhanced to let you maintain insert/replace mode across invocations.
- Dyadic ⎕*CHDIR* has been implemented to let you change the default drive.
- The session manager in the Extended System has been merged with the interpreter so you now have both parts of the system in one executable file.
- A new batch file is included to make installation easier. The file contains an APL startup template, which you can modify for your configuration. The revised *Installation Guide* also includes "quick installation" instructions.
- The obsolete U= and V= startup parameters have been eliminated.

System Requirements

The APL★PLUS System for the PC, Version 10, runs on IBM PC, XT, AT, PS/2, or compatible microcomputers with an IBM or Hercules monochrome, CGA, EGA, or VGA display adapter. One double-sided disk drive, 384K RAM memory and DOS 2.0 or later are required. A hard disk, 640K RAM, and a math co-processor are recommended. For network support, DOS 3.1 or later is required.

STSC

STSC, Inc.
2115 East Jefferson Street
Rockville, Maryland 20852
800-592-0050
301-984-5123 (in Maryland)
Outside the U.S.A., contact your local APL★PLUS dealer, or call 301-984-5412
Telefax 301-984-5290
Telex 898085 (STSCROVE)

APL★PLUS is a registered trademark of STSC, Inc. Other product names mentioned in this brochure may also be trademarks, and are used here for identification only.
Copyright © 1990, STSC, Inc.
Printed in the U.S.A.
CC268-1090